

# **UNIVERSIDAD NACIONAL DE PIURA ESCUELA DE POSGRADO**

**UNIDAD DE POSGRADO DE LA FACULTAD DE  
INGENIERÍA INDUSTRIAL**



**PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA  
INFORMACIÓN Y COMUNICACIONES**

## **TESIS**

**ANÁLISIS COMPARATIVO EN LA IMPLEMENTACIÓN DE LA RED  
NEURONAL BACKPROPAGATION USANDO EL MÉTODO DE  
COMPONENTES PRINCIPALES Y EL MÉTODO CLÁSICO**

**PARA OPTAR EL GRADO ACADÉMICO DE DOCTOR EN  
TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**

**Piura – Perú**

**Marzo, 2019**

UNIVERSIDAD NACIONAL DE PIURA

# ACTA DE SUSTENTACIÓN

**DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**

Los Miembros del Jurado Calificador que suscriben, reunidos para la sustentación de la Tesis Doctoral, para optar el Grado Académico de Doctor en **TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES** Presentada por:

AGUILAR IBAÑEZ - LUIS

Con el asesoramiento del DR. CESAR HARO DIAZ, denominada:

# “ANÁLISIS COMPARATIVO EN LA IMPLEMENTACIÓN DE LA RED NEURONAL BACKPROPAGATION USANDO EL MÉTODO DE COMPONENTES PRINCIPALES Y EL MÉTODO CLASICO”

Oídas las respuestas y absueltas las observaciones formuladas, se declara:

|                  |                      |              |                  |                    |
|------------------|----------------------|--------------|------------------|--------------------|
| <b>APROBADO</b>  |                      |              |                  | <b>DESAPROBADO</b> |
| <i>Excelente</i> | <i>Sobresaliente</i> | <i>Bueno</i> | <i>Aceptable</i> |                    |
| _____            | _____                | _____        | _____            | _____              |

En consecuencia, previa aprobación del Art.º 83, del Reglamento General de la Escuela de Posgrado, queda en condiciones de ser calificado **APTO** para obtener el Grado Académico de **DOCTOR EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**. De conformidad con lo estipulado en la ley.

PIURA, JUEVES 07 DE MARZO DEL 2019.

DR. REUCHER CORREA MOROCHO  
PRESIDENTE

DR. RAMON COSME CORREA BECERRA  
SECRETARIO

**DR. CARLOS EDUARDO CABRERA PRIETO**  
**VOCAL**

**DR. VICTOR HUGO RAMIREZ ORDINOLA**  
**VOCAL.**

DR. CARLOS ENRIQUE ARELLANO RAMIREZ  
VOCAL



# **UNIVERSIDAD NACIONAL DE PIURA ESCUELA DE POSGRADO**

**UNIDAD DE POSGRADO DE LA FACULTAD DE  
INGENIERÍA INDUSTRIAL**




**PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA  
INFORMACIÓN Y COMUNICACIONES**


## **TESIS**

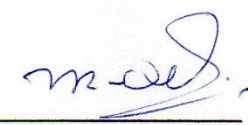
**ANÁLISIS COMPARATIVO EN LA IMPLEMENTACIÓN DE LA RED  
NEURONAL BACKPROPAGATION USANDO EL MÉTODO DE  
COMPONENTES PRINCIPALES Y EL MÉTODO CLÁSICO**

**APROBADA EN CONTENIDO Y ESTILO POR:**

  
\_\_\_\_\_  
**Dr. REUCHER CORREA MOROCHO**  
**PRESIDENTE**

  
\_\_\_\_\_  
**Dr. RAMÓN C. CORREA BECERRA**  
**SECRETARIO**

  
\_\_\_\_\_  
**Dr. CARLOS E. CABRERA PRIETO**  
**VOCAL**

  
\_\_\_\_\_  
**Dr. VÍCTOR RAMÍREZ ORDINOLA**  
**VOCAL**

  
\_\_\_\_\_  
**Dr. CARLOS E. ARELLANO RAMÍREZ**  
**VOCAL**

# **UNIVERSIDAD NACIONAL DE PIURA ESCUELA DE POSGRADO**

**UNIDAD DE POSGRADO DE LA FACULTAD  
DE INGENIERÍA INDUSTRIAL**




**PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA  
INFORMACIÓN Y COMUNICACIONES**

## **TESIS**

**ANÁLISIS COMPARATIVO EN LA IMPLEMENTACIÓN DE LA RED  
NEURONAL BACKPROPAGATION USANDO EL MÉTODO DE  
COMPONENTES PRINCIPALES Y EL MÉTODO CLÁSICO**

**LOS SUSCRITOS DECLARAMOS QUE LA PRESENTE TESIS ES ORIGINAL  
EN SU CONTENIDO Y FORMA:**

  
\_\_\_\_\_  
**Lic. Luis Aguilar Ibáñez, M.Sc**  
**EJECUTOR**

  
\_\_\_\_\_  
**Dr. Cesar Leonardo Haro Díaz**  
**ASESOR**

**© Luis Aguilar Ibáñez 2019. Todos los Derechos Reservados.**

**A la memoria de mis adorados Padres**

**María Susana Ibáñez Castro**

**y**

**Nolberto Aguilar Meza**

Agradezco a mi asesor:  
**Dr. Cesar Leonardo Haro Díaz**  
por sus sugerencias y aportes

Agradezco al Ingeniero:  
**Luis Antonio Aguilar Gutiérrez** por su invaluable  
y constante apoyo en el desarrollo de la presente tesis

Agradezco:  
A todas las personas que, con sus  
consejos, sugerencias y críticas, me  
ayudan a seguir adelante.

## RESUMEN

El objetivo principal de esta investigación fue realizar un análisis comparativo en la implementación de redes neuronales de *Backpropagation* utilizando el método multivariado PCA frente al método clásico, que en este ámbito de investigación se definió como la aplicación de redes neuronales sin la implementación de PCA.

El enfoque en esta tesis doctoral, concretamente, referente a PCA, fue aplicarlo a una variedad de conjuntos de datos. Los resultados se calcularon desde la perspectiva de las redes neuronales, incluyendo redes totalmente conectadas y convolucionales. Para este propósito, se seleccionaron tres tipos de conjuntos de datos. El primero representa datos tabulares, mientras que los otros dos datos de imagen, que están compuestos por imágenes RGB de 1 canal y 3 canales.

Las técnicas de reducción de la dimensión, como la PCA, contribuyen en gran medida al proceso de flujo de aprendizaje automático. Por ejemplo, PCA se puede usar para seleccionar características, también puede transformar conjuntos de datos en representaciones más bajas. Al mismo tiempo, PCA se puede utilizar como una herramienta para visualizar datos de alta dimensión. Además, dado que PCA reduce el número de dimensiones, también reduce los recursos computacionales y acelera el entrenamiento.

Del mismo modo, también se ha observado cómo la combinación de PCA con otros algoritmos mejora el rendimiento de clasificación. Sin embargo, los resultados de las mejoras son relativos a las diferentes configuraciones utilizadas por los modelos. Además, pueden verse afectados por factores independientes, que no están relacionados con PCA. Por lo tanto, es necesario un análisis imparcial para obtener una mejor comprensión del efecto de PCA sobre los algoritmos de aprendizaje.



En este sentido, los resultados obtenidos en esta investigación mostraron un incremento considerable en la precisión cuando se usa PCA con redes neuronales totalmente conectadas. Este hecho fue más frecuente con los conjuntos de datos tabulares.

Los mismos resultados se observaron con las imágenes de 1 canal. A pesar de que las mejoras en la precisión no fueron demasiado significativas, especialmente en comparación con otros modelos, como las redes convolucionales.

Sin embargo, cuando se aplicó PCA a las imágenes de 3 canales, con un conjunto de modelos de red pre-entrenados, la precisión no mejoró. De hecho, la precisión fue mucho menor en comparación con el modelo clásico (resnet normal).

Esto sugiere que la aplicación de PCA mejora la precisión en algunos tipos de conjuntos de datos más que en otros. En particular, cuando los datos son más complejos, PCA puede perder esa complejidad, lo que resulta en ratios de precisión más bajos.

**Palabras Clave:** *Reducción de dimensiones, redes neuronales, análisis de componentes principales, algoritmo Backpropagation, aprendizaje de máquina.*

## **ABSTRACT**

The main purpose in this research was to perform a comparative analysis in the implementation of the backpropagation neural networks using the multivariate method PCA against the classic method, which in this research scope was defined as the application of neuronal networks without the PCA implementation.

The approach in this doctoral thesis, concretely, referent to PCA, was to applied it to a variety of data sets. The results were computed from the neural networks perspective, including fully connected and convolutional networks. For this purpose, three types of data sets were selected. The first one represents tabular data, meanwhile the other two image data, which are composed by 1 channel and 3 channel RGB images.

Dimensionality reduction techniques, such as PCA, greatly contribute in the machine learning flow process. For example, PCA can be used to select features, it can also transform data sets into lower representations. At the same time, PCA can be used as a tool to visualize high dimensional data. Moreover, since PCA reduce the number of dimensions, it also reduces the computational resources and speed up the training.

In the same way, it has also been observed how the combination of PCA with other algorithms improve their classification performance. However, the results in improvements are relative to the different configurations used by the models. Moreover, they can be affected by independent factors, which are not related to PCA. Therefore, an unbiased analysis is necessary in order to obtain a better understanding in the effect of PCA over the learning algorithms.

In this regard, the results obtained in this research shown a considerable increment in accuracy when PCA is used with fully connected neural networks. This fact was more prevalent with the tabular data sets.

The same results were observed with the 1-channel images. Even though the accuracy improvements were not too significative, especially when compared with other models, such as convolutional networks.

However, when PCA was applied to the 3-channel images, with a set of pretrained resnet models, the accuracy did not improve. In fact, the accuracy was much lower in comparison with the classic model (normal resnet). This suggest that the application of PCA improves the accuracy in some types of data set more than in others. In particular, when the data is more complex, PCA can lose that complexity, which results in lower accuracy ratios.

**Keywords:** *Dimensionality reduction, neural networks, principal component analysis, Backpropagation algorithm, machine learning.*

## RESUMO

O objetivo principal desta pesquisa foi realizar uma análise comparativa na implementação das redes neurais de retropropagação utilizando o método multivariado PCA contra o método clássico, que neste escopo de pesquisa foi definido como a aplicação de redes neuronais sem a implementação do PCA.

A abordagem nesta tese de doutorado, concretamente, referente ao PCA, foi aplicá-lo a uma variedade de conjuntos de dados. Os resultados foram computados a partir da perspectiva de redes neurais, incluindo redes totalmente conectadas e convolucionais. Para este propósito, três tipos de conjuntos de dados foram selecionados. O primeiro representa dados tabulares, enquanto os outros dois dados de imagem são compostos por imagens RGB de 1 e 3 canais.

Técnicas de redução de dimensionalidade, como o PCA, contribuem muito no processo de fluxo de aprendizado de máquina. Por exemplo, o PCA pode ser usado para selecionar recursos, ele também pode transformar conjuntos de dados em representações inferiores. Ao mesmo tempo, o PCA pode ser usado como uma ferramenta para visualizar dados de alta dimensão. Além disso, como o PCA reduz o número de dimensões, também reduz os recursos computacionais e acelera o treinamento.

Da mesma forma, também foi observado como a combinação de PCA com outros algoritmos melhora seu desempenho de classificação. No entanto, os resultados em melhorias são relativos às diferentes configurações usadas pelos modelos. Além disso, eles podem ser afetados por fatores independentes, que não estão relacionados à PCA. Portanto, uma análise imparcial é necessária para obter uma melhor compreensão do efeito do PCA sobre os algoritmos de aprendizagem.

A este respeito, os resultados obtidos nesta pesquisa mostraram um

incremento considerável na precisão quando o PCA é usado com redes neurais totalmente conectadas. Esse fato foi mais prevalente nos conjuntos de dados tabulares.

Os mesmos resultados foram observados com as imagens de 1 canal. Mesmo que as melhorias de precisão não tenham sido muito significativas, especialmente quando comparadas com outros modelos, como redes convolucionais.

No entanto, quando o PCA foi aplicado às imagens de 3 canais, com um conjunto de modelos de resnet pré-tratados, a precisão não melhorou. De fato, a precisão foi muito menor em comparação com o modelo clássico (resnet normal). Isso sugere que a aplicação do PCA melhora a precisão em alguns tipos de conjuntos de dados mais do que em outros. Em particular, quando os dados são mais complexos, o PCA pode perder essa complexidade, o que resulta em menores taxas de precisão.

**Palavras-Chave:** *Redução de dimensionalidade, redes neurais, análise de componentes principais, algoritmo de retropropagação, aprendizado de máquina.*



# ÍNDICE DE CONTENIDO

|  |           |
|--|-----------|
| AGRADECIMIENTOS                                  | vi        |
| RESUMEN  | vii       |
| ABSTRACT   | ix        |
| RESUMO   | xi        |
| ÍNDICE DE TABLAS                                 | xv        |
| ÍNDICE DE FIGURAS                                | xvi       |
| <b>INTRODUCCIÓN</b>                              | <b>1</b>  |
| <b>CAPÍTULO I: EL PROBLEMA DE INVESTIGACION</b>  | <b>4</b>  |
| 1.1. PLANTEAMIENTO DEL PROBLEMA                  | 4         |
| 1.1.1. Descripción de la realidad problemática   | 4         |
| 1.1.2. El problema principal                     | 6         |
| 1.2. OBJETIVOS DE LA INVESTIGACIÓN               | 7         |
| 1.2.1. Objetivo General                          | 7         |
| 1.2.2. Objetivos Específicos                     | 7         |
| 1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN           | 7         |
| 1.4. FACTIBILIDAD DE LA INVESTIGACIÓN            | 9         |
| 1.5. DELIMITACIÓN                                | 9         |
| 1.6. LIMITACIONES                                | 10        |
| 1.7. FORMULACIÓN DE LA HIPÓTESIS                 | 10        |
| 1.7.1. Hipótesis General (HG)                    | 10        |
| 1.7.2. Hipótesis Específicas (HE)                | 10        |
| 1.7.3. Variables e Indicadores                   | 11        |
| <b>CAPÍTULO II: FUNDAMENTO TEÓRICO</b>           | <b>12</b> |
| 2.1. ANTECEDENTES DE LA INVESTIGACIÓN            | 12        |
| 2.1.1. Internacionales                           | 12        |
| 2.1.2. Nacionales                                | 19        |
| 2.1.3. Locales                                   | 23        |
| 2.2. MARCO HISTÓRICO                             | 24        |
| 2.3. BASES CIENTÍFICAS                           | 26        |
| 2.3.1. Cerebro y Redes Neuronales Artificiales   | 26        |
| 2.3.2. Análisis de Componentes Principales (PCA) | 28        |
| 2.3.3. Redes neuronales convoluciones            | 31        |
| 2.3.4. Redes neuronales backpropagation          | 34        |
| 2.3.5. Gradiente descendente                     | 37        |
| 2.3.6. Transferencia de aprendizaje              | 38        |

|   |  |           |
|---|--|-----------|
| 2.4.  | MARCO FILOSÓFICO .....                               | 41        |
| <b>CAPÍTULO III: MARCO METODOLÓGICO.....</b>            |  | <b>46</b> |
| 3.1.  | METODOLOGÍA DE INVESTIGACIÓN .....                   | 46        |
| 3.2.  | DATA SET.....  | 46        |
| 3.2.1.  | Data tabular.....                                    | 47        |
| 3.2.2.  | Imágenes-1C (1 canal).....                           | 48        |
|   | Fuente: Elaboración propia. Clases de data set ..... | 49        |
| 3.2.3.  | Imágenes-3C.....                                     | 49        |
| 3.3.  | PREPROCESAMIENTO .....                               | 50        |
| 3.4.  | DIVISIÓN DE LOS DATOS.....                           | 51        |
| 3.5.  | APLICACIÓN DE PCA.....                               | 52        |
| 3.6.  | FASE DE ENTRENAMIENTO .....                          | 52        |
| 3.6.1.  | Modelos para datos tabulares .....                   | 53        |
| 3.6.2.  | Modelos para imágenes-1C.....                        | 54        |
| 3.6.3.  | Modelos para imágenes-3C.....                        | 55        |
| 3.7.  | ANÁLISIS DE RESULTADOS.....                          | 57        |
| 3.8.  | TIPO DE INVESTIGACIÓN .....                          | 58        |
| 3.9.  | POBLACIÓN Y MUESTRA .....                            | 60        |
| 3.10.   | TÉCNICAS Y HERRAMIENTAS .....                        | 60        |
| <b>CAPÍTULO IV: RESULTADOS DE LA INVESTIGACIÓN.....</b> |  | <b>61</b> |
| 4.1.  | DATA SET TABULARES .....                             | 61        |
| 4.2.  | DATA SET DE IMÁGENES – 1C .....                      | 66        |
| 4.3.  | DATA SET DE IMÁGENES - 3C .....                      | 68        |
| 4.4.  | VALIDACIÓN DE HIPÓTESIS .....                        | 71        |
| <b>CONCLUSIONES.....</b>                                |  | <b>75</b> |
| <b>RECOMENDACIONES.....</b>                             |  | <b>77</b> |
| <b>ANEXOS.....</b>                                      |  | <b>81</b> |
| <b>ANEXO A. MATRIZ DE CONSISTENCIA.....</b>             |  | <b>82</b> |
| <b>ANEXO B. CONFIGURACIÓN DE HIPER-PARÁMETROS .....</b> |  | <b>84</b> |

## ÍNDICE DE TABLAS

| <b><u>Tabla</u></b>   | <b><u>Página</u></b> |
|---|----------------------|
| Tabla 1: Descripción de data sets tabulares. ....   | 47                   |
| Tabla 2: Descripción de la data set imágenes-1C.....  | 48                   |
| Tabla 3: Data set Imágenes-1C, lista de clases. ....  | 49                   |
| Tabla 4: Descripción de la data set imágenes-3C.....  | 49                   |
| Tabla 5: Data set Imágenes-3C, lista de clases. ....  | 49                   |
| Tabla 6: División de los data sets tabulares.....   | 51                   |
| Tabla 7: División de data set imágenes-1C.....  | 51                   |
| Tabla 8: División de data set imágenes-3C.....  | 51                   |
| Tabla 9: Descripción de modelos para data set Imágenes-3C. ....   | 56                   |
| Tabla 10: Efecto de PCA sobre los datos tabulares. ....   | 61                   |
| Tabla 11: Comparación de performance de aprendizaje entre PCA y el<br>Método Clásico en data set tabulares..... | 65                   |
| Tabla 12: Efecto de PCA sobre imágenes-1C. ....   | 66                   |
| Tabla 13: Comparación de performance de aprendizaje entre PCA y el<br>Método Clásico en imágenes-1C.....        | 68                   |
| Tabla 14: Efecto de PCA sobre imágenes-3C. ....   | 69                   |
| Tabla 15: Comparación de accuracy entre modelos pre-entrenados .....  | 71                   |
| Tabla 16: Configuración para modelos en data set tabulares.....   | 84                   |
| Tabla 17: Configuración para modelos en data set Imágenes-1C. ....  | 84                   |
| Tabla 18: Configuración para modelos en data set imágenes-3C. ....  | 85                   |
| Tabla 19: Aumentaciones usadas en data set imágenes-3C. ....  | 85                   |

## ÍNDICE DE FIGURAS

| <b><u>Figuras</u></b>  | <b><u>Página</u></b> |
|--|----------------------|
| Figura 1: Arquitectura de modelo clásico para datos tabulares.....   | 53                   |
| Figura 2: Arquitectura de modelo PCA para datos tabulares. ....  | 54                   |
| Figura 3: Arquitectura de modelo clásico para imágenes-1C. ....  | 54                   |
| Figura 4: Arquitectura de modelo PCA para imágenes-1C. ....  | 55                   |
| Figura 5: Arquitectura de modelo convolucional para imágenes-1C. ....  | 55                   |
| Figura 6: Arquitectura de modelos en data set Imágenes-3C.....   | 57                   |
| Figura 7: Vista global de la metodología. ....   | 58                   |
| Figura 8: Visualización de componentes principales para los data set<br>tabulares. ....                                | 62                   |
| Figura 9: Visualización de datos de modelo clásico (a), PCA (b),<br>convolucional (c) y reconstrucción de PCA (d)..... | 67                   |
| Figura 10: Visualización de características usando PCA (a) - (d) y datos<br>normales (e). ....                         | 70                   |

# INTRODUCCIÓN

En la última década, las Redes Neuronales Artificiales (RNA) han recibido un particular interés como una tecnología en tareas de minería de datos, aprendizaje de máquina y recientemente aprendizaje profundo. Esto sucede debido a la capacidad de abstracción de las RNA, lo cual les permite modelar de manera efectiva y eficiente problemas complejos que involucran cantidades grandes de datos.

Para que los modelos RNA puedan ser aplicados, se requiere de datos que representen la naturaleza del problema. Estos datos son usados por las RNA para encontrar patrones u características de forma inductiva por medio de un proceso denominado entrenamiento. Esta tarea puede ser realizada de manera supervisada o no supervisada.

En tal sentido, las RNA, pueden ser consideradas como un método de resolver problemas, de forma individual o combinada con otros métodos para tareas de clasificación, identificación, diagnóstico, optimización o predicción en las que el balance datos/conocimiento se inclina hacia los datos y donde, adicionalmente, puede existir la necesidad de aprendizaje en tiempo de ejecución y de cierta tolerancia a fallos. En estos casos las RNA se adaptan dinámicamente reajustando constantemente los “pesos” de sus interconexiones.

Las RNA inspiran su comportamiento y/o funcionamiento en sus contrapartes biológicas. Donde, cada neurona por separado es una unidad capaz de procesar señales (impulsos) de manera individual. Sin embargo, al combinarse entre ellas formando redes de interconexiones, la capacidad de procesamiento y/o complejidad aumentan significativamente. La neurona, recibe impulsos externos, los cuales representan sus entradas. Estos son multiplicados por valores denominados pesos; los cuales simulan la capacidad de aprendizaje de la red. El



resultado de los pesos es combinado en una suma, donde se agrega un valor de sesgo. Ahora, para simular la activación del impulso nervioso, se aplica una función de activación, la cual determina si se dispara o no la señal. Asimismo, es posible agrupar estas unidades dentro de otras estructuras denominadas capas o niveles. Estas capas son capaces de comunicarse entre ellas gracias a la interconexión que existe entre las neuronas. Asimismo, en este esquema, las señales son procesadas hacia delante, siguiendo el mismo mecanismo de procesamiento en cada neurona. A través de una estrategia de entrenamiento, la cual puede ser supervisada o no supervisada, las RNA ajustan su arquitectura y parámetros para minimizar una función de error que indique el grado de ajuste a los datos y la capacidad de generalización de las RNA.

Sin embargo, cuando los data sets son altamente dimensionales, se necesita de un elevado número de unidades en la capa de entrada para poder captar las características. Asimismo, se incrementan los recursos computacionales usados para el entrenamiento de las redes. En este escenario es posible aplicar técnicas de reducción de dimensiones. Particularmente, PCA, es capaz de reducir las dimensiones, a la vez que conserva cierto grado de varianza de los datos originales. Esto se traduce a las redes neuronales como menos nodos de entrada, por lo tanto, el entrenamiento será más rápido, a la vez que se ha eliminado características que no tienen un grado aceptable de varianza.

Es así que, en esta Tesis Doctoral, se explora los efectos de PCA sobre las redes neuronales, para lo cual se aplica PCA a diversos tipos de data sets. Para efectos comparativos, por cada data set se implementan dos tipos de redes neuronales. Una de ellas es denominada red clásica o método clásico, esta implementa la red neuronal con los datos originales. Mientras que la otra red, denominada red PCA, usa los datos reconstruidos por PCA como datos de entrada. Asimismo, se aplican diferentes tipos de redes neuronales, así como

diferentes estrategias de aprendizaje. El contenido de esta investigación se encuentra estructurado de la siguiente manera:

En el capítulo I, muestra los aspectos iniciales de la investigación, estos incluyen: Planteamiento del Problema, Justificación e Hipótesis.

El capítulo II, establece los antecedentes de la investigación, así como todo el fundamento teórico requerido para la implementación de las secciones posteriores.

El capítulo III contiene una descripción detallada de la metodología aplicada para la corroboración de las hipótesis de investigación planteadas en esta Tesis Doctoral.

El capítulo IV muestra los resultados obtenidos en la investigación. Estos representan la ejecución de la metodología propuesta en el capítulo III.

En el capítulo V se analizan los resultados obtenidos en la investigación. Asimismo, en la sección de Conclusiones, se discuten los aspectos más relevantes encontrados durante la investigación. Finalmente, sección de recomendaciones detalla algunos puntos que, a consideración del investigador pueden implementarse para ampliar la investigación.

# **CAPÍTULO I: EL PROBLEMA DE INVESTIGACION**

## **1.1. PLANTEAMIENTO DEL PROBLEMA**

### **1.1.1. Descripción de la realidad problemática**

El análisis multivariable (AM) es el conjunto de métodos estadístico-matemáticos, destinados a describir e interpretar la relación existente entre variables en un conjunto de datos. En el caso del aprendizaje automático, estos métodos son usados para identificar aquellas variables o patrones dentro de un conjunto de datos que mantienen una relevancia significativa (Zhang, 2017). De esta manera, es posible estimar la importancia para cada una de las variables existentes en un determinado conjunto de datos. Esta información es usada en diversas aplicaciones como la selección e ingeniería de características (ASonawale & Ade, 2015). Por ende, estos métodos, permiten disminuir el espacio de dimensiones, al retirar aquellas variables que son irrelevantes. En este contexto, existen diversos métodos como: el análisis de componentes principales (PCA), análisis lineal discriminante (LDA), kernel PCA (KPCA), entre otros.(Sarveniazi, 2014), diferenciándose en el criterio de selección de la importancia de las variables.

Asimismo, la aplicación de estas técnicas supone una ayuda a los investigadores al momento de analizar datos compuestos por un número elevado de dimensiones, ya que permiten seleccionar las variables más representativas. Por ejemplo, PCA, permite disminuir la información redundante y el ruido del conjunto de

datos mediante una transformación lineal que disminuye el número de variables hasta un número deseado. Logrando, de esta manera, mantener la máxima cantidad de varianza posible en el nuevo conjunto de datos.

Para la aplicación de estos conceptos, se considera un problema recurrente en el análisis e implementación de las redes neuronales de diferente tipo y en particular las redes *Backpropagation*, pero, es necesario disponer de gran cantidad de datos, para realizar su entrenamiento correspondiente. Si bien es cierto que disponer de muchos datos representa una ventaja para el entrenamiento, también incrementa la complejidad y recursos computacionales necesarios para el entrenamiento y validación.

Asimismo, tanto la interpretación como visualización del conjunto de datos se complica en presencia de un número elevado de dimensiones. Esto, debido a la dificultad para visualizar y efectuar una interpretación gráfica adecuada en presencia de más de tres variables (dimensiones).

En cuanto al entrenamiento de las RNA el algoritmo *backpropagation* (propagación hacia atrás) es usado ampliamente para la propagación de los errores usando la técnica de gradiente descendente, conocido también como: “regla delta generalizada”, debido a que supone una extensión de la regla de aprendizaje propuesta por Widrow y Hoff en 1960. Por ende, el objetivo es propagar los errores desde las unidades de salida hacia atrás, ya que, en un sistema de este tipo, el error de una unidad mantiene una relación con las unidades de salida a las que dicha unidad está conectada. Tras conocerse el error cometido por las unidades

intermedias, pueden entonces ajustarse las conexiones entre unidades de entrada y unidades intermedias. De forma similar a la regla delta, la base matemática del algoritmo *backpropagation* es la técnica de gradiente decreciente, la cual se basa en ajustar los pesos en la dirección opuesta al gradiente, esto es, en la dirección que determina el decremento del error más rápido.

El algoritmo de propagación inversa ha hecho posible el diseño de redes neuronales de múltiples capas para numerosas aplicaciones, como el control adaptativo, la clasificación la predicción del mercado de valores y el reconocimiento de voz.(Kumar, Mathew, & Swaminathan, 2010).

Dado un espacio arbitrario no se puede asegurar la convergencia, por lo que las redes neuronales y el análisis de componentes principales deben construirse en un espacio funcional donde se asegure la convergencia a un punto de dicho espacio al haberse definido antes una métrica.

### **1.1.2. El problema principal**

¿Es posible efectuar el análisis comparativo de los resultados obtenidos a partir de la implementación de la red neuronal *Backpropagation* mediante la aplicación del Análisis de Componentes Principales (PCA) y el método clásico?

Esta interrogante nos hace pensar en preguntas específicas, que guiarán el proceso de investigación:

**PE1:** ¿En qué medida se reduce la dimensionalidad de las variables aplicando el PCA en el tratamiento de datos multivariados?



**PE2:** ¿Cuál es el efecto de los resultados del PCA en la implementación de la RNA *Backpropagation*?

**PE3:** ¿En qué medida PCA mejora la representación gráfica de los datos multivariantes mediante el uso de un *framework*?

**PE4:** ¿En qué medida el uso de los resultados del PCA, influye en el entrenamiento de la RNA?

**PE5:** ¿En qué medida el hecho de que el espacio data set sea métrico garantiza la convergencia?

## **1.2. OBJETIVOS DE LA INVESTIGACIÓN**

### **1.2.1. Objetivo General**

Realizar el análisis comparativo en la implementación de la red neuronal *Backpropagation* usando el PCA y el esquema clásico.

### **1.2.2. Objetivos Específicos**

- Reducir la dimensionalidad de las variables aplicando el PCA en el tratamiento de los datos multivariantes.
- Determinar el efecto de los resultados del PCA en la Implementación de la red neuronal artificial (RNA) *Backpropagation*.
- Determinar una mejor representación gráfica de los datos multivariantes usando un *framework*.
- Determinar la influencia en el entrenamiento de la RNA usando los resultados del PCA.
- Determinar si el espacio data set es métrico.

## **1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN**

Cuando se dispone de muchas variables, el proceso de aprendizaje de la

red neuronal consume gran cantidad de recursos y tiempo. La implementación del proceso del PCA implica la simplificación de variables a partir de lo cual el entrenamiento de la red neuronal resulta mucho más rápido y no se incrementa el error de convergencia significativamente. Por tanto, los resultados no pierden su validez.

La representación geométrica con menos variables permite un mejor análisis respecto a la rapidez de la convergencia de los datos. Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Las redes neuronales convoluciones constan de múltiples capas de filtros convolucionales de una o más dimensiones. Después de cada capa, por lo general se añade una función para realizar un mapeo causal no-lineal. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Se espera que con este Modelo las RNA obtengan ventajas, que incluyen:

- Aprendizaje Adaptativo: capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- Autoorganización: una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- Tolerancia a fallos: la destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.

- Operación en tiempo real: los cómputos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
- Fácil inserción dentro de la tecnología existente: se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

Asimismo, este tipo de tecnología permitirá abrir espacio dentro de una de las ramas de la Inteligencia Artificial, y se desarrolle en el Perú, como una propuesta de solución a múltiples problemas o necesidades que demanden de realizar pronósticos, predicciones, estimaciones, entre otras.

#### **1.4. FACTIBILIDAD DE LA INVESTIGACIÓN**

La factibilidad, se trata de una propuesta de acción para resolver el problema de investigación. Para llevar a cabo el proyecto factible, lo primero que se realizó es el diagnóstico de la situación planteada; en segundo lugar, se ha planteado y fundamentado con basamentos teóricos la propuesta a elaborar y establecer, tanto los procedimientos metodológicos, así como las actividades y los recursos necesarios, para llevar a delante la ejecución. Se ha dispuesto de todos los recursos necesarios para lograr los objetivos y contrastación de la hipótesis en base a los resultados obtenidos.

#### **1.5. DELIMITACIÓN**

El ámbito de aplicación de este modelo es para todo tipo de organización, e investigadores a nivel nacional e internacional preocupados en el

desarrollo de las redes neuronales y sus aplicaciones.

## **1.6. LIMITACIONES**

Para esta investigación se está trabajando solamente con tres tipos de data sets. Estos corresponden a datos tabulares, imágenes de un solo canal e imágenes de tres canales. Asimismo, todos los modelos empleados corresponden al modelo de redes neuronales. Finalmente, el proceso de entrenamiento de los modelos se encuentra relacionado con el hardware usado: GPU: Tesla K80, CPU: Xeon Processors, RAM: 13GB.

## **1.7. FORMULACIÓN DE LA HIPÓTESIS**

### **1.7.1. Hipótesis General (HG)**

Existe una diferencia significativa en la implementación de la red neuronal *backpropagation* usando los resultados del PCA y del método clásico.

### **1.7.2. Hipótesis Específicas (HE)**

**HE1:** Aplicando el PCA en el tratamiento de datos multivariantes se reduce la dimensionalidad de las variables.

**HE2:** Los resultados del PCA influyen en la implementación de la RNA.

**HE3:** PCA, mejora la representación geométrica de los datos obtenidos mediante el uso de un framework.

**HE4:** Los resultados del PCA influyen en el entrenamiento de la RNA.

**HE5:** La métrica definida sobre el espacio data set garantiza la convergencia.

### 1.7.3. Variables e Indicadores

#### **Variable Independiente**

Análisis de Componentes Principales (PCA), método clásico.

#### **Indicadores**

- Número de Variables al final de la implementación del PCA.
- Métrica de performance *Accuracy*.

#### **Variable dependiente**

Análisis comparativo en la implementación de la red neuronal

#### **Indicadores**

- Número de dimensiones.
- Número de vectores de entrada.
- Número de componentes de principales.
- Promedio de Métrica de performance *Accuracy*
- Número de iteraciones
- Convergencia.

#### **Instrumentos**

- Lenguaje de programación Python.
- Librería Pytorch.
- Librería Scikit-learn.
- Redes totalmente conectadas.
- Redes convoluciones.
- Redes pre-entrenadas resnet18.
- Visualizaciones.
- Sucesiones.

## CAPÍTULO II: FUNDAMENTO TEÓRICO

### 2.1. ANTECEDENTES DE LA INVESTIGACIÓN

#### 2.1.1. Internacionales

##### Antecedente 1

(Bravo, 2007) En esta tesis doctoral elaborada en la Universidad de Alcalá titulada **“Arquitectura basada en FPGAs para la detección de objetos en movimiento, utilizando visión computacional y técnicas PCA”**, se plantea una nueva propuesta para la detección de objetos en movimiento dentro de una escena, mediante la plataforma de propósito específico construida en base a una FPGA (Field Programmable Gate Array) de la capacitación y gestión de las imágenes en blanco y negro procedentes de una CMOS, de resolución 1.2 M píxeles, realiza la detección de nuevos objetos presentes en la escena mediante la aplicación de la técnica PCA (Análisis de Componentes principales).

Los problemas más importantes que se plantean en esta tesis son los relacionados con la implementación de la técnica PCA de forma íntegra en una FPGA. El elevado número de operaciones que demanda la técnica PCA conllevan elevados tiempos de ejecución si se usan plataformas de procesamiento convencionales, por esta razón se usan técnicas PCA para hardware reconfigurable.

Para ello se ha paralizado la ejecución de las diferentes fases que forman el algoritmo PCA: Cálculo de la matriz de covarianza y cálculo de autovalores, proyección y recuperación de imágenes al y desde el espacio transformado y la detección de nuevos objetos. Con la solución propuesta e implementada en esta tesis se la

conseguido un sistema segmentado que alcanza un alto ratio de imágenes procesado por el segundo (Aproximadamente 120 imágenes). Dentro del algoritmo PCA se realizaron 4 tipos diferentes de multiplicación de matrices (cálculo de matriz de covarianza, obtención de la matriz de transformación proyección y recuperación desde el espacio transformado), por lo que es necesario la implementación de una nueva propuesta para la multiplicación de matrices FPGA. Concretamente se construyó un arreglo semi-sistólico de propósito específico, el cual posee una estructura modular que permite su reutilización de los distintos productos de matrices que requiere la técnica PCA. Aunque las dimensiones de las matrices a multiplicar son diferentes debido a la estructura regular del arreglo, éste adapta su ejecución en función del producto a realizar.

El cálculo de autovalores y autovectores son las operaciones que demandan mayor carga computacional dentro de la técnica del PCA, condicionando el tiempo total consumido en la ejecución del algoritmo del PCA. Para resolver estos elementos en el FPGA se desarrolló un método basado en el método de Jacobi. La nueva propuesta consume pocos recursos internos del FPGA en comparación con otras alternativas.

Los resultados reales alcanzados son óptimos ya que además de alcanzar un alto ratio de imágenes procesadas. El sistema desarrollado logra una alta eficiencia a la hora de detectar nuevos objetos dentro de la escena, así en el peor de los casos el número de falsos positivos (detección de un nuevo objeto cuando no lo hay) asciende a 17% y de los falsos negativos (omisión de detección de

un nuevo objeto) a un 9%. En el mejor de los casos el índice de falsos positivos y negativos es menor a 1%.

## **Antecedente 2**

**(Palacios, 2014)** En esta tesis doctoral titulada “**Aprendizaje Digital y Morfología Matemática aplicados al Procesamiento de Imágenes Biomédicas**” elaborada en la Universidad Nacional de Mar del Plata. Se propone un nuevo paradigma para el diseño automático de operadores morfológicos invariantes ante traslaciones y localmente definidos por medio de una ventana, llamado W-operadores. El paradigma propuesto consiste en definir y representar a un W-operador para clasificación y segmentación mediante un sistema de reconocimiento de patrones. El resultado de analizar una imagen es un conjunto de características o atributos que representan dicha imagen. El procesamiento y el análisis de imágenes se basan en el uso de funciones lineales y no lineales llamadas operadores de imágenes y transformadas respectivamente.

El método empieza analizando la morfología matemática que es una técnica no lineal de procesamiento y análisis de imágenes. Su función es la caracterización geométrica de las mismas y el análisis de texturas basado en la teoría de conjuntos. el procedimiento a seguir para un problema determinado usualmente consiste en probar varias secuencias de operaciones morfológicas sobre un conjunto de imágenes de diseño. Este conjunto contiene imágenes que son una muestra del problema a resolver. Las secuencias de operaciones a probar se evalúan en función de alguna medida de



error que refleja el objetivo del diseño. Luego, una vez que el diseñador ha encontrado una secuencia apropiada de operaciones, entre todas las secuencias testeadas. En la etapa de testeo se procede a estimar la capacidad predictiva de dicha secuencia usando otro conjunto de imágenes, denominado conjunto de imágenes de testeo.

El algoritmo de aprendizaje se selecciona teniendo en cuenta la cantidad de datos disponibles para el diseño, entre otras características del problema en cuestión. La estimación de la función característica se realiza de tal modo que, el costo de estimación o diseño sea el más bajo posible, lo cual asegura que el error del operador diseñado sea también bajo. En la etapa de testeo se estima el error del operador diseñado mediante el cálculo del error empírico.

Este error permite evaluar la capacidad predictiva, o futuro desempeño, del operador diseñado cuando se aplica a nuevas imágenes del problema en cuestión que no fueron registradas en los conjuntos de entrenamiento y testeo.

El diseño del clasificador se realiza mediante la regla *plug-in* y una regresión no lineal entre el cociente de las probabilidades de clase de las configuraciones de ventana y la respuesta de una red neuronal tipo *feedforward* de tres capas. Para el entrenamiento de la red neuronal se utiliza como función de costo al error cuadrático medio ponderado. En la primera parte de esta tesis se propone el diseño de W-operadores binarios como un problema de diseño de clasificadores binarios. Luego se presenta un breve resumen del diseño de W-operadores usando la regla *plug-in* considerando el

problema de generalización. En base a este análisis y teoría se presenta de manera formal el método propuesto para el diseño de W-operadores binarios.

Luego se abordan aspectos concernientes a su implementación práctica. Finalmente, se presentan cuatro experimentos donde uno de ellos involucra el procesamiento de imágenes médicas y los tres restantes son problemas generales de PDI. El problema de imágenes médicas consiste en el filtrado de ruido en la segmentación de vasos sanguíneos en imágenes oculares.

Los problemas generales de PDI aquí considerados son la detección de bordes en imágenes contaminadas con ruido artificial, identificación de texturas en imágenes de mapas, y reconocimiento de caracteres en imágenes de texto. En base a estos experimentos se analizan y comparan los resultados obtenidos con los resultados provenientes de la aplicación de operadores de imágenes diseñados usando la regla  $k$ NN, multiresolución piramidal, máquinas de soporte vectorial, gradiente morfológico, y redes neuronales convolucionales, se propone un nuevo método para el diseño automático de W-operadores en el procesamiento de imágenes a color.

Estas imágenes se caracterizan por el hecho de que cada píxel toma un vector de tres valores y no un escalar como en las imágenes en escala de grises o binarias. En este contexto, se considera el problema particular donde las imágenes de entrada del procesamiento son imágenes color y las imágenes de salida son imágenes binarias. En función de esto se define a un W-operador mediante un clasificador binario. Para las imágenes color, y un

tamaño de ventana dado, el número de componentes de una configuración de ventana es tres veces más grande que para el caso de imágenes en escala de grises o binarias.

Esto implica un crecimiento exponencial del tamaño del espacio de búsqueda del clasificador o W-operador óptimo. A nivel práctico, para una cantidad fija y limitada de ejemplos de entrenamiento, esta situación obliga a definir restricciones para reducir el costo de estimación y el costo computacional de diseño de clasificadores.

### **Antecedente 3**

**(Faik, 2016)** En esta tesis postdoctoral titulada **“Face recognition in low-resolution images under small sample conditions with face-part detection alignment”**, desarrollada en la universidad de Gromingen. El autor empieza definiendo tres objetivos para la investigación. Estos tienen que ver con localización de caras, alineación y la identificación. Aunque este campo tiene mucho estudio, quedan retos aún por trabajar para el reconocimiento facial.

Tal es el caso de iluminación, variaciones, caras de similitudes dentro de una familia, así como posibles dificultades para encontrar suficientes muestras de entrenamiento (que también deben tener suficiente resolución). De obtener los datos en bruto para proporcionarlos a un algoritmo de reconocimiento facial algunos procesos son obligatorios, como ubicar la cara en la imagen de una cámara y alinearla para evitar sufrir ruidos debidos a variaciones de postura e iluminación. Por todos estos hechos, el primer objetivo fue localizar la cara con precisión. Esto se hace justo después de la detección de la cara. Después del marco facial se obtiene del

detector facial, se desarrolló un nuevo algoritmo detector de par de ojos como un elemento importante ubicación facial en una cara.

A veces el número de muestras disponibles no es más que unas pocas. Esto se llama el problema de la muestra pequeña (SSP). En algunos casos, hay incluso solo una muestra por materia. En el caso extremo del SSP también conocido. como la muestra única por persona; problema (SSPP). El objetivo fue encontrar una solución a estos problemas.

Para trabajar el problema del SSPP, se usó dos algoritmos nuevos que varían jerárquicamente. Estos son el algoritmo de selección de región basado en máxima similitud (MSRS) y el método de cálculo de distancia basado en Multi-HOG. Mientras que el primer método intenta encontrar las regiones más similares para evitar recortar y plantear errores, el segundo utiliza una función de cálculo de distancia basada en múltiples HOG para obtener finalmente las identidades de la cara. Para manejar el problema del SSP, el autor usó un método de bolsa de palabras basado en HOG (HOG-BOW). En el método HOG-BOW, un algoritmo de agrupación de k-medio construye un libro de códigos entrenado en muchos parches extraídos en la etapa de entrenamiento. Después de la construcción del libro de códigos, se utiliza para crear vectores de características por medio de un método de asignación suave. Los resultados de ambos métodos mostraron una mejora significativa para el reconocimiento facial en los casos de SSP y SSPP.

Los modelos basados en la apariencia generan un modelo a partir de imágenes de los ojos usando la apariencia fotométrica de ojos. Dado que no hay información específica a priori relacionado

con los ojos, se necesitan parámetros para la detección del ojo, con el fin de eliminar ruido y reduciendo dimensionalidad, extracción de características y normalización.

Usualmente se aplican operaciones a datos de entrenamiento. En cuanto a las técnicas de extracción de características, se usa análisis de componentes principales (PCA) y detección de bordes los métodos son algunas de las técnicas utilizadas. Después de todas estas operaciones la salida se lleva a un clasificador para el entrenamiento como clasificadores, adaptativos, impulso, se usan redes neuronales y SVM (máquinas de soporte vectorial). Los resultados son altamente satisfactorios para los objetivos de la investigación.

### 2.1.2. Nacionales

#### **Antecedente 1**

**(Vega, 2011)** En esta tesis Doctoral realizada en la Universidad Nacional San Marcos titulada **“Redes neuronales para el reconocimiento de la calidad morfológica de mangos exportables para la empresa Biofruit del Perú S.A.C. ”**, el autor utiliza un software basado en el proceso de imágenes digitales de mangos mediante redes neuronales artificiales. Primero se selecciona una muestra de mangos exportables; luego, utilizando el sistema, tomamos una imagen digital de cada mango, con dichas imágenes entrenamos una Red Neuronal para que reconozca acertadamente todos los elementos de la muestra. Luego, se procede con las pruebas de reconocimiento con otros mangos de calidad y con los que no presentan dichas características.

Esta investigación tiene dos etapas bien diferenciadas, en la primera etapa se ha empleado un diseño no experimental y en la segunda etapa un diseño experimental. La etapa no experimental corresponde al manejo de los valores de las formas de los mangos y su calidad exportable de los mismos, estos son tomados de la naturaleza como tales y no pueden ser manipuladas por el investigador. En esta etapa se elabora un modelo matemático para la captura del patrón morfológico de los mangos basado las imágenes de los mismos capturados mediante cámaras digitales. Para la segunda etapa, el modelo el diseño experimental que se usó es el modelo de prueba y posprueba con grupos experimentales.

El Sistema Experto basado en RNA, se concluye que la empresa Biofruit puede realizar la clasificación de mangos exportables con un margen de error de 2.33%. El margen de error del personal de clasificación de mangos es de 17.3%, lo que quiere decir que se ha logrado una optimización del 15 %. Respecto al patrón morfológico de un mango, concluimos que el contorno o borde de su imagen, generado por el software, es un patrón apropiado para entrenar adecuadamente una RNA. El diseño de la RNA que ha sido capaz de aprender a reconocer los 30 mangos de la muestra tiene la siguiente configuración: 2500 neuronas en la capa de entrada (para tomar como entrada de la RNA el borde de la imagen de un mango), 3 neuronas en la capa de salida (para que la RNA pueda identificar tres tipos de mangos exportables) y 400 neuronas en la capa oculta (para que pueda realizar un entrenamiento adecuado).

## **Antecedente 2**

**(Peña, 2016)** En esta tesis doctoral desarrollada en la Universidad Nacional Agraria la Molina, titulada “ **Uso de Redes Neuronales Artificiales para Optimizar la Dosificación de Coagulantes en la Planta de Tratamiento de Agua Potable en Huancayo**” En la investigación se utilizaron las redes neuronales artificiales como herramienta para optimizar la cantidad de coagulante ( $Al_2SO_4$ ) que se utiliza en el tratamiento del agua potable de la empresa SEDAM de la ciudad de Huancayo, donde la calidad del agua al ingreso a la planta de tratamiento tuvo variabilidad permanente de turbidez debido a las precipitaciones fluviales que se presentaron generalmente en los meses de octubre a marzo.

Este fenómeno meteorológico alteró en corto tiempo las características físicas y químicas del agua dificultando su tratamiento. Se utilizó en el laboratorio el ensayo denominado Prueba de Jarras para determinar la dosis de coagulante a emplear durante un periodo de once meses, con muestreo aleatorio diario.

Las variables consideradas para el estudio fueron la turbidez, el pH, la conductividad, el color y sólidos disueltos totales con una temperatura promedio de 10 °C y un caudal promedio de 148 l/s. Los cálculos obtenidos fueron utilizados para entrenar y probar dosis óptimas de coagulante haciendo uso de redes neuronales artificiales con tres topologías específicas: Red de tres capas (entrada de cinco neuronas, oculta de cinco neuronas, salida una neurona), Red de cuatro capas (entrada de cinco neuronas, dos ocultas de seis y cinco neuronas, salida una neurona) y Red de

cuatro capas con dos ocultas llamada también de regresión generalizada (una de entrada con cinco neuronas, dos capas ocultas, en la primera se adiciona una neurona por cada caso entrenado y en la segunda oculta dos neuronas receptoras para regresión y una capa de salida con una neurona).

Se concluyó que la topología más adecuada fue la última con un asertividad de 96.9 por ciento frente a los datos reales y en su entrenamiento se obtuvo una correlación de 98.4 por ciento evidenciando además que los factores más influentes al modelo son: color, turbidez y pH.

La red seleccionada finalmente permitió predecir la dosificación de coagulante óptimo con una probabilidad de error máximo de 1.6 por ciento y en tiempo real con nuevos valores de entrada en el agua a tratar sin necesidad de recurrir a la Prueba de Jarras el cual sólo se realiza posteriormente para enriquecer su aprendizaje.

Según la evaluación de las configuraciones de redes propuestas; la Red Neuronal Artificial de Regresión Generalizada (GRN) presenta mejor rendimiento en sus resultados de entrenamiento y prueba de la dosis óptima de coagulante de sulfato de aluminio en el tratamiento de agua potable aplicando distintos parámetros físicos y químicos de entrada, siendo el porcentaje de error de 3,73 por ciento en el entrenamiento y su índice de determinación en el entrenamiento de 96,9 por ciento, lo cual nos indica una correlación de 98,4 por ciento cercana al 100 por ciento entre el pronóstico de la red neuronal y las pruebas del laboratorio reales.

La predicción de dosificación de coagulante calculada en tiempo



real por la configuración de la Red Neuronal Artificial de Regresión Generalizada (GRN) permite tomar decisiones rápidas y con errores menores al cuatro por ciento ante cambios inesperados de turbiedad y/o pH que se puedan presentar cuando el afluente se altera debido a las precipitaciones fluviales intempestivas que caen en la zona.

### 2.1.3. Locales

#### Antecedente 1

(Ojeda, 2016) En la tesis doctoral, desarrollada en la Universidad Nacional de Piura y titulada “**Reconocimiento de Patrones Basado en Redes Neuronales Artificiales**” se resuelve el problema de reconocimiento de patrones, haciendo uso de Redes Neuronales Artificiales, realizando el diseño e implementación de patrones digitales.

El modelo se construyó después de identificar diferentes algoritmos de aprendizaje para reconocimiento de patrones. Se implementó una red neuronal artificial en un ambiente de lenguaje de programación JAVA, con el uso del *framework* JOONE.

La implementación de la red neuronal, permite el reconocimiento de patrones con un error permisible del 2%. Los resultados muestran que el error en el aprendizaje disminuye cada vez más rápido. Se describen los aspectos metodológicos en fases y actividades, donde destaca de preparación de la data set, el entrenamiento de la red neuronal y finalmente el reconocimiento de patrones. El diseño del modelo de la RN se construyó después de identificar los diferentes algoritmos de aprendizaje para el

reconocimiento de patrones.

La red se implementó con patrones digitales que luego se transforman en matrices de valores 0 y 1, luego estas matrices se transforman en vectores que luego son procesadas por la red.

## **2.2. MARCO HISTÓRICO**

Haciendo un análisis del momento histórico de las redes neuronales como elemento de la IA (**Lake, Ullman, Tenenbaum, & Gershman, 2017**) en el artículo “Building machines that learn and think like people “ afirma:

La inteligencia artificial (IA) ha tenido una historia de auges y caídas, sin embargo, para cualquier medida tradicional de éxito, los últimos años han estado marcados por un progreso excepcional. Gran parte de este progreso provienen de los avances recientes en "aprendizaje profundo", caracterizados por el aprendizaje de grandes estilos de redes neuronales, modelos con múltiples capas de representación. Estos modelos han logrado notables avances en muchos dominios que abarcan reconocimiento de objetos, reconocimiento de voz y control (LeCun, Bengio, y Hinton, 2015); (Schmidhuber, 2015). En reconocimiento de objetos, Krizhevsky, Sutskever y Hinton. (2012) entrenó una red neuronal convolucional profunda (convnets; LeCun et al., 1989) que casi redujo a la mitad la tasa de error del estado de la técnica anterior en el punto de referencia más difícil hasta la fecha.

En los años posteriores, las redes continúan dominando, acercándose recientemente al rendimiento a nivel humano en algunos puntos de referencia de reconocimiento de objetos (He, Zhang, Ren y Sun, 2015; Russakovsky et al., 2015; Szegedy et al., 2014). En el reconocimiento de voz automático, los modelos ocultos de Markov (HMM) ha sido el enfoque

principal desde fines de la década de 1980 (Juang y Rabiner, 1990), pero este marco se ha desgarrado pieza por pieza y se ha reemplazado con componentes de aprendizaje profundo (Hinton et al., 2012). Ahora, los enfoques principales para el reconocimiento de voz son sistemas de redes completamente neuronales (Graves, Mohamed, & Hinton, 2013; Weng, Yu, Watanabe y Juang, 2014). Las ideas de aprendizaje profundo también se han aplicado al aprendizaje de problemas de control complejos. V. Mnih et al. (2015) combinaron ideas de aprendizaje profundo y aprendizaje de refuerzo para hacer un algoritmo de "aprendizaje de refuerzo profundo" que aprende a jugar grandes clases de videojuegos simples desde solo fotogramas de píxeles y la puntuación del juego, lograr un rendimiento a nivel humano o sobrehumano en muchos de estos juegos (ver también Guo, Singh, Lee, Lewis y Wang, 2014; Schaul, Quan, Antonoglou y Plata, 2016; Stadie, Levine, & Abbeel, 2016).

Estos logros han ayudado a las redes neuronales a recuperar su estatus de paradigma líder en aprendizaje automático, como lo fueron a finales de los 80 y principios de los 90. El reciente éxito de Las redes neuronales ha llamado la atención más allá de la academia. En la industria, empresas como Google. y Facebook tienen divisiones de investigación activas que exploran estas tecnologías, en el objeto y el habla. Los sistemas de reconocimiento basados en el aprendizaje profundo se han implementado en productos principales en teléfonos inteligentes. y la web. Los medios de comunicación también han cubierto muchos de los logros recientes de las redes neuronales, a menudo expresando la opinión de que las redes neuronales han logrado este éxito reciente en virtud de la computación similar al cerebro y, por lo tanto, su capacidad para emular el aprendizaje humano y la cognición humana.

Durante casi todo el tiempo que han existido redes neuronales, ha ocurrido críticas a las redes neuronales. (Crick, 1989; Fodor y Pylyshyn, 1988; Marcus, 1998, 2001; Minsky y Papert, 1969; Pinker & Prince, 1988). Sin embargo, es indudable su utilidad en muchos tipos de problemas de aprendizaje automático, que demuestran el poder del aprendizaje basado en gradientes y Jerarquías profundas de variables latentes. Las redes neuronales también tienen una rica historia como modelos computacionales de cognición (McClelland, Rumelhart y el Grupo de Investigación de PDP, 1986; Rumelhart, McClelland, y el Grupo de Investigación PDP, 1986).

En un nivel más fundamental, cualquier modelo computacional de aprendizaje debe, en última instancia, estar cimentado en las redes neuronales biológicas del cerebro.

## **2.3. BASES CIENTÍFICAS**

### **2.3.1. Cerebro y Redes Neuronales Artificiales**

Las RNA tienen su inspiración en el funcionamiento del cerebro. Según **(Gurney, 2004)** en el artículo titulado “ Neural Networks”, considera:

Una red neuronal es un conjunto interconectado de elementos simples de procesamiento, unidades o nodos, cuya funcionalidad está basada en la neurona animal. La capacidad de procesamiento de la red son las fortalezas de conexión entre unidades, o pesos, obtenidos por un proceso de adaptación o aprendizaje de un conjunto de patrones de entrenamiento.

El cerebro humano consta de un estimado de 1011 (100 mil millones) de células nerviosas o neuronas. Las neuronas se

comunican vía señales eléctricas que son impulsos de corta duración o "picos" de voltaje de la pared celular o membrana.

Las conexiones interneurónicas están mediadas por uniones electroquímicas llamadas sinapsis, que se encuentran en las ramas de la célula denominadas dendritas. Cada neurona normalmente recibe muchos miles de conexiones de otras neuronas y, por lo tanto, recibe constantemente una multitud de señales entrantes, que finalmente llegan al cuerpo celular. Aquí, están integrados o sumados de alguna manera y, en términos generales, si la señal resultante supera algún umbral, la neurona "disparará" o generará un impulso de voltaje en respuesta. Esto luego se transmite a otras neuronas a través de una fibra ramificada conocida como el axón.

Al determinar si se debe producir un impulso o no, algunas señales entrantes producen un efecto inhibitorio para evitar disparos, mientras que otros son excitadores y promueven la generación de impulsos.

En cuanto a la importancia de las conexiones interneuronales, este tipo de sistema a veces se denomina conexionista y el estudio de este enfoque general como conexionismo. Esta terminología es a menudo la que se encuentra en las redes neuronales en el contexto de modelos psicológicamente inspirados de la función cognitiva humana.

Las señales ponderadas se suman juntas mediante una simple suma aritmética para suministrar una activación de nodo, la llamada unidad lógica de umbral. La activación se compara con un

umbral; si la activación excede el umbral, la unidad produce una salida de alto valor (convencionalmente "1"), de lo contrario, arroja cero.

Los nodos están dispuestos en una estructura en capas en la que cada señal emana de una entrada y pasa a través de dos nodos antes de llegar a una salida más allá de la cual ya no se transforma. Esta estructura *feedforward* es solo una de varias disponibles y típicamente se utiliza para colocar un patrón de entrada en una de varias clases según el patrón resultante de las salidas

La neurona funciona como un dispositivo de procesamiento de señales. Esta capacidad se debe en gran parte a las propiedades de la membrana o el recubrimiento de la superficie de la neurona, que admite una amplia variedad de procesos electroquímicos. Morfológicamente, la principal diferencia radica en el conjunto de fibras que emanan del cuerpo celular. Una de estas fibras, el axón, es responsable de transmitir señales a otras neuronas y, por lo tanto, puede considerarse la salida de neuronas.

### **2.3.2. Análisis de Componentes Principales (PCA)**

Respecto a éste interesante tema seguimos lo mostrado por **(Lippi & Ceccarelli, 2019)** en el artículo "Exact implementation and continuity correction" quien afirma y define: El análisis de componentes principales (PCA) es una técnica utilizada y un tema muy estudiado en la literatura. PCA es una técnica para reducir la dimensionalidad de los datos de un conjunto de variables correlacionadas.

Esta técnica se implementa tradicionalmente como una operación por lotes fuera de línea.

El concepto de continuidad está estrictamente relacionado con la naturaleza incremental del algoritmo propuesto: en PCA estándar, el análisis de lotes implica que la noción de tiempo no existe, por ejemplo. El orden de los elementos en el conjunto de muestra no es relevante para el algoritmo de lote. El cálculo para el PCA comienza considerando un conjunto de datos observados.

$$X_n = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

Donde  $x_i$  es un vector fila de longitud “m”, mientras que  $X_n$  representa una matriz real de orden  $n \times m$ . El siguiente paso es definir las medias muestrales  $\bar{x}_n$  y la desviación estándar  $\sigma_n$  con respecto a las columnas, escritas en forma habitual como:

$$\bar{x}_{n(j)} = \frac{1}{n} \sum_{i=1}^n X_{n(ij)}$$

$$\sigma_{n(j)} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n [X_{n(ij)} - \bar{x}_{n(j)}]^2}$$

Donde entre paréntesis escribimos los índices matriciales y vectoriales explícitamente, “j” representa las columnas. De esta manera podemos definir la matriz estandarizada para los datos como:

$$Z_n = \begin{bmatrix} x_1 - \bar{x}_n \\ x_2 - \bar{x}_n \\ x_3 - \bar{x}_n \\ \vdots \\ x_n - \bar{x}_n \end{bmatrix} \sum_n^{-1}$$

Donde  $\sum_n = \text{diag}(\sigma_n)$  es una matriz  $m \times n$ . Luego se define la matriz de covarianza  $Q_n$  de la matriz de datos  $X_n$  como:

$$Q_n = \frac{1}{n-1} Z_n^T Z_n$$

Vemos que  $Q_n$  es para cualquier  $n$  una matriz  $m \times m$  simétrica y definida positiva. Finalmente hacemos una diagonalización estándar para que podemos escribir:

$$Q_n = C_n^{-1} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} C_n$$

Donde los valores propios (positivos)  $\lambda_i$  están en orden descendente:  $\lambda_i > \lambda_{i+1}$ . La matriz de transformación  $C_n$  es la matriz de vectores propios y es ortogonal,  $C_n^{-1} = C_n^T$ . Sus filas son los componentes principales de la matriz  $Q_n$ . y el valor de  $\lambda_i$  representa la varianza asociada al  $i^{\text{th}}$  componente principal. Ajustando  $P_n = Z_n C_n$ , tenemos una evolución temporal para los valores de las PCs hasta paso de tiempo  $n$ .

Recordamos que el procedimiento de diagonalización no está definido de manera única: una vez que se elige el orden de los valores propios, todavía se puede elegir el "signo" del vector



propio para los espacios propios de una dimensión y una base ortonormal adecuada para los degenerados. Hacemos hincapié que, dado que solo la estructura de espacios *eigenspaces* es una propiedad intrínseca de los datos, las PCs son útiles en cantidad para su interpretación, pero no están definidos de forma única.

### 2.3.3. Redes neuronales convoluciones

Con respecto a este tipo de redes neuronales (**Awate, Bangare, Pradeepini, & Patil, 2018**) en el artículo "Detection of Alzheimer's Disease from MRI using Convolutional Neural Network with Tensorflow" afirma:

Las redes neuronales convolucionales (CNN) se utilizan principalmente para datos gráficos en reconocimiento y procesamiento de imágenes. Están inspiradas en Los sistemas visuales humanos y fueron creados por Yann LeCun et al (1998).

La imagen de entrada se descompone. en trozos más pequeños conocidos como campos receptivos locales. CNN viene con 3 capas principales cada una de ellas tiene una función específica para realizar, a saber, capa convolucional, capa de agrupación y capa totalmente conectada. La capa convolucional (aquí denominada CL) tiene un conjunto de filtros de aprendizaje que se pueden aplicar en imágenes de entrada. También consta de 3 dimensiones y p espacial y porciones espaciales. Cada operación requiere 4 hiperparámetros:

$K$ , número de filtros; Vector de pesos a aplicar a la imagen de entrada.

$F$ , su extensión espacial; 3 Dimensiones: ancho, alto y profundidad.

$S$ , el tamaño de zancada; Cantidad de desplazamiento del filtro, en píxeles.

$P$ , la cantidad de (cero) relleno; Se controla la entrada desproporcionada para adaptarse al filtro.

La aplicación de un filtro o una operación de convolución implica desplazamiento, escala y rotación. Otras transformaciones para encontrar patrones, CNN toma imágenes de entrada como  $W_1 * H_1 * D_1$ .

Luego, la imagen se define como:

$$W_2 = \frac{(W_1 - F)}{S} + 1$$

$$H_2 = \frac{(H - F)}{S} + 1$$

$$D_2 = D_1$$

Mientras que CL maneja la filtración de imágenes, la capa de agrupación (aquí denominada PL) se ocupa de reducir el tamaño de la muestra y el control de sobreajuste. Normalmente se coloca entre CL y capa totalmente conectada. (Después de referido como FCL); que es una red neuronal clásica con cada nodo interconectado con cualquier otro nodo. La unidad lineal rectificadora (denominada aquí como RELU) se utiliza para obtener los segmentos sin firmar en el espacio de entrada. En CNN los

usaremos como nuestra función de activación.

Las RELU también tienen una ventaja distinguible; que no requiere que se normalice la entrada que evita que se saturan. En el artículo "Supervised learning in image classification" (J. He & Xu, 2019), afirma:

Una imagen bidimensional que generalmente está representada por un tensor, para clasificar una colección de imágenes en " $k$ " clases distintivas.

$$f \in D := \mathbb{R}^{m \times n \times c}$$

$$\text{Donde } c = \begin{cases} 1 & \text{para imágenes en blanco y negro} \\ 3 & \text{para imágenes a color} \end{cases}$$

Un problema típico de aprendizaje automático supervisado comienza con un conjunto de datos (datos de entrenamiento)

$$D = \{(f_i, y_i)\}_{i=1}^N$$

$$\{f_i\}_{i=1}^N \subset D$$

$y_i \in \mathbb{R}^k$  es la etiqueta para datos  $f_i$  con  $[y_i]_j$  como la probabilidad para  $f_i$  en las clases " $j$ ".

En términos generales, un problema de aprendizaje supervisado puede considerarse como un problema de ajuste de datos en un nivel de espacio dimensional alto  $D$ . A saber, necesitamos encontrar un mapeo.

$$H: \mathbb{R}^{m \times n \times c} \rightarrow \mathbb{R}^k$$

Tal que para cada  $f \in D$ ,  $H(f) \approx e_i \in \mathbb{R}^k$ .

Si  $f$  está en la clase  $i$ , para  $1 \leq i \leq \kappa$ . Para la configuración general anterior, utilizamos un modelo probabilístico, para entendiendo la salida  $H(f) \in \mathbb{R}^k$  como una distribución discreta en  $\{1, 2, \dots, k\}$  con  $[H(f)]_i$  como la probabilidad de  $f$  en la clase  $i$ , esto es:

$$0 \leq [H(f)]_i \leq 1, \quad \sum_{i=1}^k [H(f)]_i = 1$$

Por fin, terminamos nuestro modelo con una estrategia simple para elegir.

$$\arg \max_i \{[H(f)]_i : i = 1:k\}$$

El paso principal en la construcción de  $H$  es construir un mapeo no lineal  $H_0: D \rightarrow V_j$ , donde:

$$V_j = \mathbb{R}^{m_j \times n_j \times c_j}$$

El subíndice  $j$  se refiere al número de vigas de engrosamiento en CNN. En términos generales, el mapa  $H_0$  desempeña dos roles. El primer papel es hacer una reducción de dimensión, a saber:

$$m_j n_j c_j \leq m n c$$

El segundo rol es mapear un conjunto complicado de datos en un conjunto de datos que son linealmente separables.

#### 2.3.4. Redes neuronales backpropagation

Respecto a este algoritmo de propagación hacia atrás (**Sáenz, A & Martín, 2002**) en el texto “Redes Neuronales y Sistemas Difusos”

afirma:

Este algoritmo ha permitido diseñar redes neuronales multicapa para numerosas aplicaciones, como control adaptativo, clasificación, predicción, reconocimiento de patrones y reconocimiento de voz. Además, las redes neuronales *backpropagation* (BPNN) tiene la ventaja de una respuesta rápida y alta precisión de aprendizaje.

Para la implementación de esta red seguimos a (Conway, 1990). Sea un MLP de tres capas, con las entradas, salidas, pesos y umbrales de las neuronas definidas. Dado un patrón de entrada  $X^\mu$  ( $\mu = 1, \dots, p$ ), recordemos que la operación global de esta arquitectura se expresa del siguiente modo.

$$z_k^\mu = \sum_j w'_{kj} y_j^\mu - \theta'_k = \sum_j w'_{kj} f\left(\sum_i w_{ji} x_i^\mu - \theta_j\right) - \theta'_k$$

Las funciones de activación de las neuronas ocultas  $f(h)$  son de tipo sigmoideo, con  $h$  el potencial postsináptico o local. La función coste de la que se parte es el error cuadrático medio.

$$E(w_{ji}, \theta_j, w'_{kj}, \theta'_k) = \frac{1}{2} \sum_\mu \sum_k \left[ t_k^\mu - f\left(\sum_j w'_{kj} y_j^\mu - \theta'_k\right) \right]^2$$

La minimización se lleva a cabo mediante descenso por el gradiente, pero en esta ocasión habrá un gradiente respecto de los pesos de la capa de salida y otro respecto de la capa oculta.

$$\delta'_{kj} = -\epsilon \frac{\partial E}{\partial w'_{kj}} \quad \delta'_{ji} = -\epsilon \frac{\partial E}{\partial w_{ji}}$$

Las expresiones de actualización de los pesos se obtienen sólo

con derivar, teniendo en cuenta las dependencias funcionales y aplicando la regla de la cadena.

$$\delta w'_{kj} = \epsilon \sum_{\mu} \Delta'_k{}^{\mu} y_j^{\mu} \quad \text{con} \quad \Delta'_k{}^{\mu} = [t_k^{\mu} - f(v_k^{\mu})] \frac{\partial f(v_k^{\mu})}{\partial v_k^{\mu}}$$

$$\delta w'_{ji} = \epsilon \sum_{\mu} \Delta'_j{}^{\mu} y_i^{\mu} \quad \text{con} \quad \Delta'_j{}^{\mu} = [\sum_k \Delta'_k{}^{\mu} w'_{kj}] \frac{\partial f(v_j^{\mu})}{\partial v_j^{\mu}}$$

La actualización de umbrales (bias) se realiza haciendo uso de estas mismas expresiones, considerando que el umbral es un caso particular de peso sináptico, cuya constante es igual a -1.

En estas expresiones está implícito el concepto de error hacia atrás de los errores que da nombre al algoritmo. En primer lugar, se calcula el algoritmo  $\Delta'_k{}^{\mu}$  que se llama **señal de error**, por ser proporcional al error de salida actual de la red, con el que calculamos la actualización  $\delta w'_{kj}$  de los pesos de la capa de salida.

A continuación, se **propagan hacia atrás** los errores  $\Delta'_k{}^{\mu}$  a través de la sinapsis, proporcionando así las señales de error  $\Delta'_j{}^{\mu}$  correspondientes a la sinapsis de la capa oculta, con estas se calcula la actualización  $\delta w_{ji}$  de las sinapsis ocultas.

En resumen, el **procedimiento** a seguir para entrenar mediante BP una arquitectura MLP dada es el siguiente:

- Establecer aleatoriamente los pesos y umbrales iniciales ( $t=0$ ).
- Para cada patrón  $\mu$  del conjunto de aprendizaje
  - 1) Llevar a cabo una fase de ejecución para obtener la respuesta de la red ante el patrón  $\mu$ -ésimo.

- 2) Calcular las señales de error asociadas  $\Delta_k'^\mu$  y  $\Delta_j'^\mu$
  - 3) Calcular el incremento parcial de los pesos y umbrales debidos a cada patrón  $\mu$  (elemento de los sumatorios)
- Calcular el incremento total (para todos los patrones) actual de los pesos  $\delta w_{kj}'$  y  $\delta w_{ji}'$ . Hacer lo mismo para los umbrales.
  - Actualizar pesos y umbrales.
  - Calcular el error actual,  $t = t+1$ , y volver a ejecutar si todavía no es satisfactorio.

Se debe comenzar siempre con **pesos iniciales** aleatorios (normalmente números pequeños, positivos y negativos), ya que si se parte de pesos y umbrales iniciales nulos el aprendizaje no progresará (puesto que las salidas de las neuronas y el incremento en los pesos serán siempre cero).

### 2.3.5. Gradiente descendente

(Sáenz, A & Martin, 2002) en el texto “Redes Neuronales y Sistemas Difusos” explica este algoritmo en los siguientes términos:

Inicialmente se eligen valores aleatorios para los pesos. La idea de este método es usar un proceso iterativo que minimice la función de error  $E(w) = \frac{1}{2} \sum_{i,p} (b_i^p - \hat{b}_i^p)^2$ . En el caso lineal, es

decir cuando  $f(x) = x$  se tiene  $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta (b_i^p - \hat{b}_i^p) a_j^p$ .

En general  $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta (b_i^p - \hat{b}_i^p) f(B_i^p) a_j^p$ . El parámetro  $\eta$

se llama tasa de aprendizaje. En el caso de funciones signoidales se tiene:

$$f(x) = \frac{1}{1+e^{-cx}} \Rightarrow f'(x) = cf(x)(1-f(x))$$

$$f(x) = \tanh(cx) \Rightarrow f'(x) = c(x)(1-f(x)^2)$$

El Perceptrón es un tipo de red de aprendizaje supervisado, es decir necesita conocer los valores esperados para cada una de las entradas presentadas; su comportamiento está definido por pares de esta forma:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}$$

Cuando  $p$  es aplicado a la red, la salida de la red es comparada con el valor esperado  $t$ , y la salida de la red está determinada por:

$$a = f\left(\sum_i w_i p_i\right) = \text{hard lim } s\left(\sum_i w_i p_i\right)$$

Los valores de los pesos determinan el funcionamiento de la red, estos valores se pueden fijar o adoptar utilizando diferentes algoritmos de entrenamiento de la red.

### 2.3.6. Transferencia de aprendizaje

(Awate et al., 2018) dice al respecto: El aprendizaje por transferencia (aquí denominado en adelante TL) es un proceso de utilización de un modelo de ML que se entrena para diferentes propósitos en diferentes datos. TL hace esto volviendo a entrenar el modelo congelado en los datos que deseamos trabajar en lo que hay que notar aquí es que solo el FL y la capa final están optimizados para los nuevos datos, no todo el modelo. Así que es muy probable que el modelo en algunas ocasiones se pierda y dar un resultado incorrecto.



### **2.3.7. Deep learning (aprendizaje profundo)**

Respecto a esta metodología (Lecun, Bengio, & Hinton, 2015) afirma: El aprendizaje por representación es un conjunto de métodos que permite a una máquina recibir información sin procesar y descubrir automáticamente las representaciones necesarias para la detección o clasificación. Los métodos de aprendizaje profundo son métodos de aprendizaje de representación con múltiples niveles de representación, obtenidos mediante la composición de módulos simples, pero no lineales que transforman la representación en un nivel (comenzando con la entrada en bruto) en una representación en un nivel superior, ligeramente más abstracto. Con la composición de suficientes transformaciones de este tipo, se pueden aprender funciones muy complejas. Para las tareas de clasificación, las capas más altas de representación amplifican aspectos de la entrada que son importantes para la discriminación y suprimen las variaciones irrelevantes. Una imagen, por ejemplo, viene en forma de una matriz de valores de píxeles, y las características aprendidas en la primera capa de representación representan típicamente la presencia o ausencia de bordes en orientaciones particulares y ubicaciones en la imagen.

La segunda capa típicamente detecta razones al detectar arreglos particulares de bordes, independientemente de pequeñas variaciones en las posiciones de los bordes. La tercera capa puede ensamblar razones en combinaciones más grandes que correspondan a partes de objetos familiares, y las capas

subsiguientes detectarían objetos como combinaciones de estas partes. El aspecto clave del aprendizaje profundo es que estas capas de características no están diseñadas por ingenieros humanos: se aprenden de los datos utilizando un procedimiento de aprendizaje de propósito general.

El aprendizaje profundo está logrando avances importantes en la solución de problemas que han resistido los mejores intentos de la comunidad de inteligencia artificial durante muchos años. Ha resultado ser muy bueno descubriendo ser visto como una especie de paisaje montañoso en el espacio de valores de pesos de alta dimensión. El vector de gradiente negativo indica la dirección del descenso más pronunciado en este paisaje, acercándolo a un mínimo, donde el error de salida es bajo en promedio.

### 2.3.8. Espacio métrico

El comportamiento de una red se modeliza mediante un espacio  $n$ -dimensional en el que cada punto representa un estado de la red, y mediante un sistema de ecuaciones diferenciales autónomo, que permite conocer la evolución de la red gracias al plano de fases (familia de trayectorias del sistema).

Sea  $X$  un espacio producto vectorial, definimos la **norma** sobre  $X$  mediante la aplicación  $\| \cdot \|: X \rightarrow R_0^+$  que cumple:

$$N1) \|x\| \geq 0; \forall x \in X$$

$$N2) \|x\| = 0 \Leftrightarrow x = 0; \forall x \in X$$

$$N3) \|\lambda x\| = |\lambda| \|x\|; \lambda \in F; \forall x \in X$$

$$N4) \|x + y\| \leq \|x\| + \|y\|; \forall x, y \in X$$

Sea  $X$  un espacio normado.  $X$  es un espacio métrico si se define una aplicación no negativa  $d: X \times X \rightarrow R_0^+$  llamada **métrica** o **distancia** que cumple las siguientes condiciones:

$$d_1: d(x, y) \geq 0$$

$$d_2: d(x, y) = 0 \Leftrightarrow x = y$$

$$d_3: d(x, y) = d(y, x)$$

$$d_4: d(x, z) \leq d(x, y) + d(y, z); \forall x, y, z \in X$$

Un espacio métrico es el par  $(X, d)$ , donde  $X$  es un espacio vectorial y " $d$ " una norma sobre  $X$ . Un espacio de Banach es un espacio normado que es completo respecto a la métrica definida por la norma. (Conway, 1990).

## 2.4. MARCO FILOSÓFICO

Tanto las redes neuronales de diverso tipo como como el PCA, son elementos de la IA. En tal sentido presentamos el marco filosófico de la IA. Al respecto (Castrillón, O. & Rodríguez, M. & Leyton, 2008) afirma:

¿Bajo qué condiciones los agentes artificiales tendrían sus propias necesidades o, a pesar de los aportes existentes, la IA aún no tiene unos límites éticos claros? Por tanto, es necesario ahondar en el estudio de este tema y acordar unos lineamientos o códigos de conducta que sean seguidos por desarrolladores, productores y consumidores. De esta manera seremos más conscientes de las consecuencias de nuestros actos y podremos establecer un contrato social o compromiso con una sociedad urgida de acciones éticas. Preguntas como ¿cuál deber ser el comportamiento ético que deban seguir los robots, máquinas y sistemas

automatizados contruidos en el futuro?, pueden generar ya sus propias conclusiones sobre la necesidad de incorporar desde el diseño de estas máquinas un componente ético y no sólo un programa que emule ciertos comportamientos o funciones humanas.

Pero este compromiso a quienes involucra: ¿A los desarrolladores de tecnología?, ¿a los consumidores?, ¿a los gobernantes? o a toda la sociedad. A opinión de (Conitzer, 2016). El comportamiento que consideramos inteligente puede replicarse en ¿una maquina? Si es así, ¿qué consecuencias podría tener esto para la sociedad? Y qué ¿Nos habla de nosotros como seres humanos? Además de ser un tema de larga data de reflexión filosófica y ciencia ficción, la IA es también un área de investigación científica bien establecida. Las hazañas logradas en tales investigaciones han sido generalmente mucho más modestas que las imaginadas en las películas. Pero con el tiempo, la brecha entre la realidad y la ficción se ha ido cerrando, los autos autodirigidos son ahora una realidad. Las oportunidades comerciales son infinitas.

En la conferencia del 2016 en Phoenix, un orador fue más controvertido que cualquier otro en la memoria reciente: Nick Bostrom. Mientras que la audiencia estaba compuesta principalmente por científicos informáticos, Bostrom es un filósofo que dirige el Instituto de Futuro de la Humanidad en Oxford. Recientemente hizo olas con su libro Superinteligencia. En él, contempla el problema que pronto podremos desarrollar una inteligencia artificial que exceda ampliamente las capacidades humanas, y considera qué pasos podemos tomar ahora para garantizar que el resultado sea lo mejor para nosotros. Una preocupación clave es la de una "explosión de inteligencia": si somos lo suficientemente inteligentes como para construir una máquina más inteligente que nosotros mismos, entonces, de acuerdo

con su pensamiento, seguramente esa máquina a su vez sería capaz de construir algo aún más inteligente, y así sucesivamente.

La frase "singularidad tecnológica" también se usa a veces para describir una inteligencia tan fuera de control. De hacerlo ¿La humanidad quedará hecha polvo? ¿Seremos eliminados? Desde la aparición del libro de Bostrom, figuras públicas como Elon Musk, Stephen Hawking y Bill Gates advirtieron sobre los riesgos de la inteligencia artificial superinteligente. Musk incluso donó \$ 10M al *Future of Life Institute* con sede en Boston, estableciendo un programa de subvenciones para asegurar que la inteligencia artificial siga siendo beneficiosa.

El reciente Director de Tecnología de la Información de las Naciones Unidas, Atefeh Riazi, se unió al coro enfatizando los riesgos de la IA. Estas preocupaciones en su mayoría han sido planteadas por personas ajenas a la comunidad de investigación en IA. Después de la charla de Bostrom, vi a varias personas expresar su disgusto en las redes sociales, diciendo que se da a ese foro le da credibilidad que no merece. Otros enfatizaron la mentalidad abierta, pero (por lo que vi) no se llegó a respaldar sus ideas. Pero supongo que la mayoría en la comunidad de inteligencia artificial se encogió de hombros y continuó con su investigación como de costumbre. ¿Por qué? ¿A los investigadores de IA simplemente no les importa el futuro de la humanidad?

Los investigadores de IA anhelaban que su trabajo fuera científicamente riguroso y respetado, y aprendieron a ser cuidadosos. Algunos trataron de disociarse del término "IA" por completo y en su lugar se asociaron con problemas técnicos más estrechamente definidos. Por ejemplo, muchos investigadores en la comunidad de aprendizaje automático, que se enfoca en que las computadoras aprendan automáticamente de los datos sobre

cómo hacer predicciones y decisiones, ya no querían ser considerados investigadores de "IA". Incluso la mayoría de los investigadores que se apegaron al término comenzaron a enfocarse en problemas más estrechos, no solo por problemas de percepción sino también por razones técnicas, pero no eran fáciles de resolver. Además, el progreso en su solución a menudo condujo a un impacto beneficioso directo en la sociedad. Por ejemplo, parte de la comunidad se ha centrado en los sistemas automatizados de planificación y programación, que se han utilizado en diversas aplicaciones.

En un panel después de la charla de Bostrom, Oren Etzioni, director del Instituto Allen de Inteligencia Artificial, provocó risas de apoyo de parte de multitud cuando señaló que la charla de Bostrom estaba felizmente carente de datos, a pesar de que Etzioni reconoció rápidamente que esto era inherente al problema. Tom Dietterich, profesor de informática en el estado de Oregón y presidente de la Asociación para el Avance de la Inteligencia Artificial expresó su escepticismo de que se produciría una explosión de inteligencia del tipo que describe Bostrom y preguntó qué experimentos podríamos realizar para probar esta hipótesis. La comunidad de IA generalmente evita las especulaciones sobre el futuro profundo y se siente más cómoda al enfrentar problemas importantes que son más concretos y tangibles en este punto, como las armas autónomas, sistemas de armas que pueden actuar sin intervención humana, o el desempleo causado por la IA que reemplaza a los trabajadores humanos.

Muchos de nosotros estamos impulsados a hacer del mundo un lugar mejor, reduciendo el número de muertes por accidentes automovilísticos, aumentando el acceso a la educación, mejorando la sostenibilidad y la asistencia sanitaria, previniendo ataques terroristas, etc.

Desafortunadamente, todavía tenemos que descubrir cómo comprometernos de manera rigurosa y productiva con los problemas filosóficos a largo plazo más nebulosos. Un área en la que parece posible cierta tracción inmediata es el estudio de cómo la inteligencia artificial (previa a la superinteligencia) puede tomar decisiones éticas, por ejemplo, cuando un automóvil que se conduce por sí mismo necesita tomar una decisión en un escenario que probablemente mate o hiera a alguien. De hecho, automatizado la toma de decisiones éticas. Este punto no está claro para los investigadores de IA cómo abordar de manera útil la noción de superinteligencia y las cuestiones filosóficas que plantea.

## **CAPÍTULO III: MARCO METODOLÓGICO**

### **3.1. METODOLOGÍA DE INVESTIGACIÓN**

Con el objetivo de poder evaluar la hipótesis planteada, y objetivos de investigación, se requiere de la creación de una metodología personalizada. Esta, debe satisfacer los requerimientos antes mencionados. En ese sentido, se optó por diseñar una metodología que involucre tanto diversos tipos de datos como redes neuronales sea posible. Ya que, de este modo, es posible realizar un análisis más profundo sobre el impacto de PCA sobre las redes neuronales. Por ello, la metodología propuesta cuenta con un total de 6 fases, estas se pueden resumir en:

- Adquisición y tipo de data sets a utilizar. Como se mencionó anteriormente, se puso especial énfasis en contar con diversos tipos de data sets.
- Preprocesamiento, estas son acciones necesarias para transformar los datos antes de ser usados por los modelos.
- División de la data set en los conjuntos de entrenamiento, validación y test respectivamente.
- Transformación de los datos usando PCA.
- Entrenamiento de los modelos, y,
- Análisis de resultados.

En los siguientes apartados se dará una descripción más detallada de las funciones de cada una de estas fases.

### **3.2. DATA SET**

Para poder implementar los algoritmos de redes neuronales es necesario contar con data sets. En ese sentido, se optó por usar diferentes tipos de



data sets. Estos, se caracterizan por presentar diferentes niveles de complejidad, número de características e instancias, tipo de clasificación, distribución de clases, etc. De tal modo que, tomando estos data sets se pueda realizar un análisis comparativo de las redes neuronales desde una perspectiva más heterogénea. En este sentido, es importante notar que, a medida que se van presentando nuevos data sets, estos aumentan sus niveles de complejidad. Es así, que se seleccionaron tres tipos de data sets.

### 3.2.1. Data tabular

Estos data sets corresponden al tipo de datos que se puede agrupar fácilmente por una matriz  $X$ , la cual contiene  $m$  instancias y  $n$  características. Asimismo, estos vienen acompañados por un vector  $y$ , el cual agrupa las clases que se desean predecir o clasificar. En concreto, para la data-tabular se reunió un total de 15 data sets, los cuales forman parte del repositorio PMLB (Olson, La Cava, Orzechowski, Urbanowicz, & Moore, 2017). En la Tabla 1 se muestra una descripción más detallada de los data sets seleccionados.

*Tabla 1: Descripción de data sets tabulares.*

| <b>Data Set</b>   | <b>Características</b> | <b>Instancias</b> | <b>N. Clases</b> |
|---|------------------------|-------------------|------------------|
| GAMETES_Epistasis_2-<br>Way_1000atts_0.4H_EDM-<br>1_EDM-1_1 | 1000                   | 1600              | 2                |
| agaricus-lepiota  | 22                     | 8145              |                  |
| GAMETES_Epistasis_2-<br>Way_20atts_0.1H_EDM-1_1             | 20                     | 1600              |                  |
| GAMETES_Epistasis_2-<br>Way_20atts_0.4H_EDM-1_1             | 20                     | 1600              |                  |
| Working on<br>GAMETES_Epistasis_3-                          | 20                     | 1600              |                  |

|                            |    |      |
|----------------------------|----|------|
| Way_20atts_0.2H_EDM-1_1    |    |      |
| GAMETES_Heterogeneity_2    |    |      |
| 0atts_1600_Het_0.4_0.2_50  | 20 | 1600 |
| _EDM-2_001                 |    |      |
| GAMETES_Heterogeneity_2    |    |      |
| 0atts_1600_Het_0.4_0.2_75  | 20 | 1600 |
| _EDM-2_001                 |    |      |
| buggyCrx                   | 15 | 690  |
| australian data            | 14 | 690  |
| analcata_data_fraud        | 11 | 42   |
| analcata_data_cyyoung8092  | 10 | 97   |
| analcata_data_japansolvent | 9  | 52   |
| analcata_data_creditscore  | 6  | 100  |
| analcata_data_aids         | 4  | 50   |
| analcata_data_lawsuit      | 4  | 264  |

Fuente: Elaboración propia. Data set tabulares

Como se puede apreciar, todos estos data sets corresponden al tipo de clasificación binaria.

### 3.2.2. Imágenes-1C (1 canal)

Este tipo de data set agrupa imágenes de un solo color o canal (1C). En concreto se decidió usar el data set *Fashion-MNIST* (Xiao, Rasul, & Vollgraf, 2017). Este data set agrupa diversas imágenes de tipos de ropa y calzado. En la Tabla 2 se detalla la descripción de la data set, mientras que en la Tabla 3 se muestran las clases.

Tabla 2: Descripción de la data set imágenes-1C.

| Data Set      | Características | Instancias | N. Clases |
|---------------|-----------------|------------|-----------|
| Fashion-MNIST | 28 x 28         | 70000      | 10        |

Fuente: Elaboración propia. Imágenes de tamaño 28 x 28

Tabla 3: Data set Imágenes-1C, lista de clases.

| Fashion-Mnist Clases |            |
|----------------------|------------|
| T-shirt/top          | Sandal     |
| Trouser              | Shirt      |
| Pullover             | Sneaker    |
| Dress                | Bag        |
| Coat                 | Ankle boot |

Fuente: Elaboración propia. Clases de data set

### 3.2.3. Imágenes-3C

Este tipo de data set agrupa imágenes RGB de 3 canales. En particular, se decidió usar el data set *flower* (Nilsback & Zisserman, 2008), el cual se compone principalmente de imágenes de diferentes especies de flores, presentando un total de 102 clases. Sin embargo, para esta investigación, se decidió seleccionar un sub-grupo de 50 clases. Los detalles de este data set se pueden observar en la *Tabla 4*.

Tabla 4: Descripción de la data set imágenes-3C

| Data Set | Características | Instancias | N. Clases |
|----------|-----------------|------------|-----------|
| Flower   | 226 x 226       | 5148       | 50        |

Fuente: Elaboración propia, Tamaño de imágenes a color

Asimismo, la Tabla 5 muestra la lista de clases seleccionadas.

Tabla 5: Data set Imágenes-3C, lista de clases.

| Flower Clases Seleccionadas |                    |
|-----------------------------|--------------------|
| grape hyacinth              | poinsettia         |
| toad Lily                   | californian poppy  |
| spear thistle               | anthurium          |
| king protea                 | morning glory      |
| oxeye Daisy                 | cape flower        |
| cautleya spicata            | bishop of llandaff |

|                           |                    |
|---------------------------|--------------------|
| Silverbush                | pink-yellow dahlia |
| black-eyed susan          | clematis           |
| Windflower                | thorn apple        |
| great masterwort          | barbeton daisy     |
| tree mallow               | bougainvillea      |
| pincushion flower         | sword lily         |
| hard-leaved pocket orchid | hibiscus           |
| bee balm                  | lotus lotus        |
| Gaura                     | cyclamen           |
| Marigold                  | foxglove           |
| ruby-lipped cattleya      | frangipani         |
| Artichoke                 | rose               |
| Gazania                   | watercress         |
| peruvian Lily             | water lily         |
| bird of paradise          | wallflower         |
| sweet William             | passion flower     |
| purple coneflower         | petunia            |
| wild pansy                | common dandelion   |
| colt's foot               | fritillary         |

---

Fuente: Elaboración propia. Clases de data set

### 3.3. PREPROCESAMIENTO

Antes de poder usar los diversos data sets, es necesario realizar tareas de preprocesamiento. Estas aseguran que el modelo reciba datos uniformes, disminuyendo de esta manera las variaciones. Para tal fin, existen diversas técnicas que pueden ser implementadas. Para el caso de esta investigación se aplica un proceso de escalamiento, el cual se define como:

$$z = (x - u)/s$$

Donde  $x$  representa una muestra de la data set,  $u$  es la media y  $s$  la desviación estándar. De esta manera, aseguramos que se remueva la media, a la vez que se escala a la varianza unitaria. Finalmente,  $z$

representa los nuevos datos obtenidos. Este esquema es aplicado para todos los tipos de data sets usados en la investigación.

### 3.4. DIVISIÓN DE LOS DATOS

En principio, se busca que los modelos puedan generalizar, es decir, que puedan realizar predicciones con datos no vistos. Para ello, es necesario realizar una división de los data sets en tres grupos:

- **Entrenamiento:** Esta parte de los datos, se usa para entrenar los modelos.
- **Validación:** Estos datos son usados para la selección de modelos.
- **Test:** Estos datos son usados para medir la capacidad de generalización de los modelos por medio de diferentes métricas de aprendizaje.

En las siguientes tablas se muestra la división en porcentajes que se ha asignado a los diferentes tipos de data sets en esta investigación.

*Tabla 6: División de los data sets tabulares.*

| <b>Entrenamiento (%)</b> | <b>Desarrollo (%)</b> | <b>Test (%)</b> |
|--------------------------|-----------------------|-----------------|
| 80.00                    | 10.00                 | 10.00           |

Fuente: Elaboración propia

*Tabla 7: División de data set imágenes-1C*

| <b>Entrenamiento (%)</b> | <b>Desarrollo (%)</b> | <b>Test (%)</b> |
|--------------------------|-----------------------|-----------------|
| 65.70                    | 20.00                 | 14.30           |

Fuente: Elaboración propia

*Tabla 8: División de data set imágenes-3C.*

| <b>Entrenamiento (%)</b> | <b>Desarrollo (%)</b> | <b>Test (%)</b> |
|--------------------------|-----------------------|-----------------|
| 72.00                    | 18.00                 | 10.00           |

Fuente: Elaboración propia

### 3.5. APLICACIÓN DE PCA

En esta fase se aplica PCA sobre los diversos data sets. El objetivo es encontrar un número adecuado de componentes principales tal que:

$$var \geq 90\%$$

Esto significa que se está perdiendo un total del 10% de varianza en los datos, lo cual se considera una pérdida aceptable dentro del marco de esta investigación. Asimismo, en esta fase se generan tres nuevos conjuntos de datos: entrenamiento, validación y test respectivamente. Estos, conjuntamente con los datos pre procesados serán usados por los modelos de redes neuronales en la siguiente fase.

### 3.6. FASE DE ENTRENAMIENTO

En esta fase se procede a entrenar los modelos de redes neuronales. Es importante notar que, en esta fase, ingresan dos tipos de data sets:

- **Originales:** Es decir, son los datos que han pasado por las fases anteriores, pero sin la aplicación de PCA.
- **Transformados:** Estos, al igual que los datos originales han pasado por todas las fases de la metodología. Sin embargo, han sido transformados usando los componentes principales, manteniendo una varianza del 90%, de acuerdo a la fase de aplicación de PCA.

Asimismo, tanto los datos originales, como los transformados se encuentran respectivamente divididos en los grupos de: entrenamiento, validación y test, de acuerdo a lo especificado en la fase de DIVISIÓN DE LOS DATOS

Ahora bien, ya que se desea realizar un análisis comparativo, es necesario entrenar dos tipos de redes. Estas difieren en el tipo de data set con el que se entrenan. Es así, que un grupo se entrenara con los datos originales (método clásico), mientras que el otro con los datos generados por PCA. Este proceso, se repite por cada tipo de data set. Asimismo, para aislar otros componentes que puedan afectar los resultados, se optó por usar una misma arquitectura para cada red. Del mismo modo se aplicó la misma configuración de hiperparámetros, esta puede ser revisada a más detalle en el ANEXO B. CONFIGURACIÓN DE HIPER-PARÁMETROS . A continuación, se detallarán los diferentes modelos usados por cada tipo de data set.

### 3.6.1. Modelos para datos tabulares

Para este tipo de data sets, se entrenaron dos modelos de redes neuronales. El primero de ellos corresponde al método clásico. En esta arquitectura, la entrada está compuesta por los datos pre-procesados. En la Figura 1 se puede observar a más detalle la implementación de esta arquitectura.

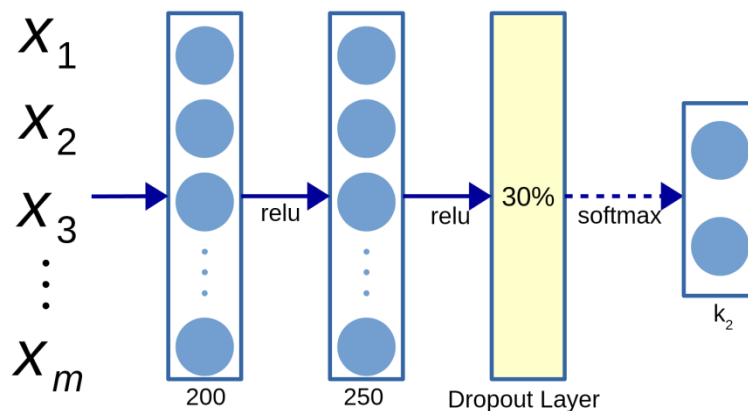


Figura 1: Arquitectura de modelo clásico para datos tabulares.

Asimismo, el segundo modelo, corresponde al modelo PCA. Esta arquitectura usa los datos generados por PCA como entradas. La Figura 2 muestra los detalles de la arquitectura.

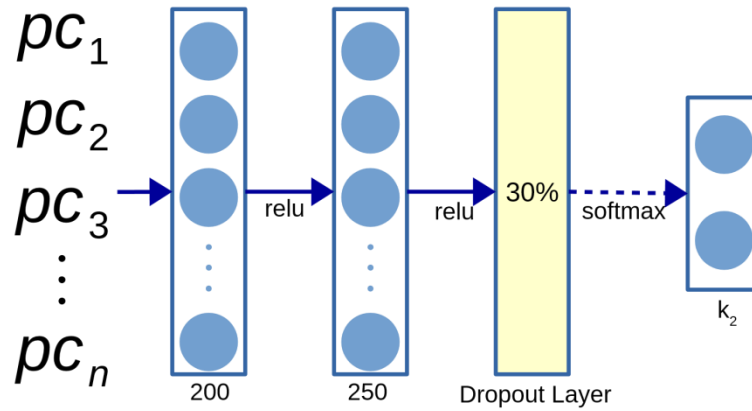


Figura 2: Arquitectura de modelo PCA para datos tabulares.

### 3.6.2. Modelos para imágenes-1C

Para este tipo de data set, se implementaron tres modelos. El primero de ellos, corresponde al modelo usando el método clásico. En este modelo, se toman como entradas las imágenes en su forma vectorial. Los detalles de la arquitectura se muestran en la Figura 3.

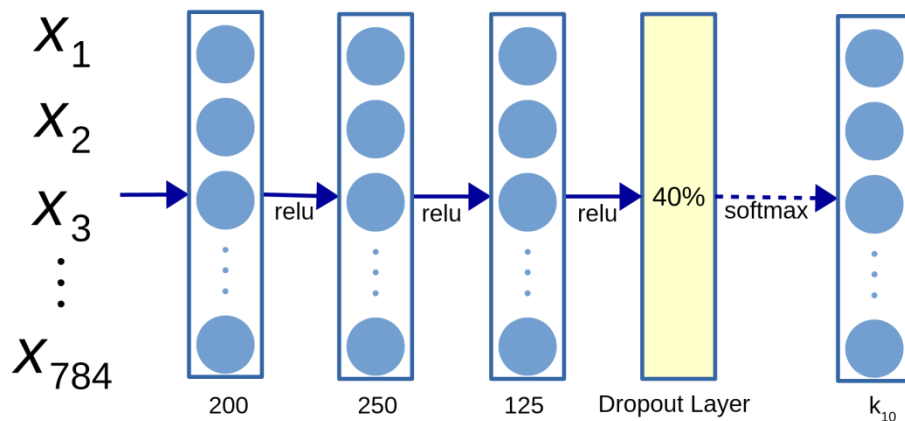


Figura 3: Arquitectura de modelo clásico para imágenes-1C.

El segundo modelo corresponde a la implementación de PCA.



Este modelo usa como entradas los datos transformados por PCA en su forma vectorial. La Figura 4 muestra la descripción de esta arquitectura.

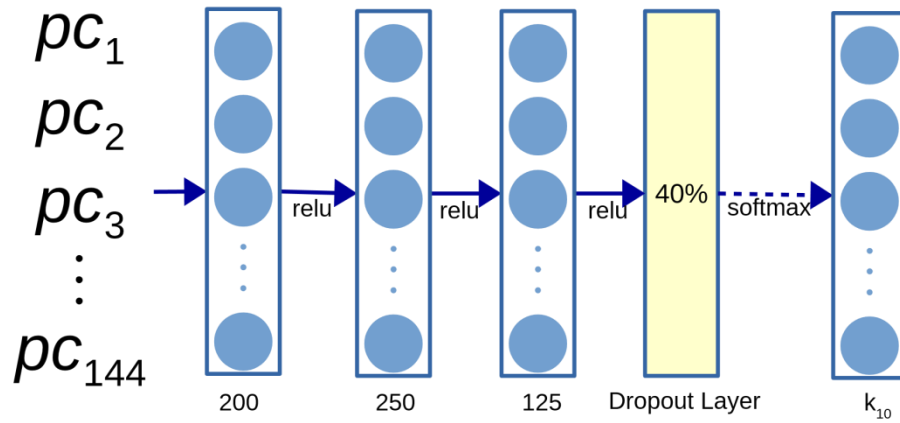


Figura 4: Arquitectura de modelo PCA para imágenes-1C.

El tercer modelo, corresponde a una red convolucional. Este modelo, a diferencia de los dos anteriores, se caracteriza por trabajar directamente con las imágenes. Es decir, no necesita transformarlos en su representación vectorial. La arquitectura de este modelo se puede observar en la Figura 5.

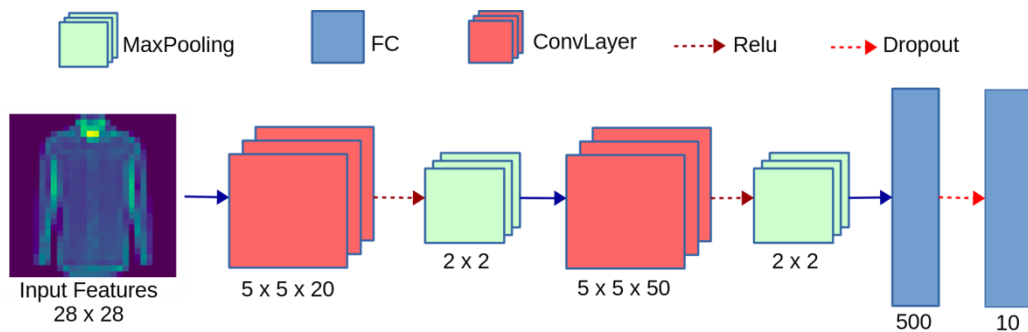


Figura 5: Arquitectura de modelo convolucional para imágenes-1C.

### 3.6.3. Modelos para imágenes-3C

Para este tipo de data set, se decidió aplicar redes convolucionales. Sin embargo, debido a la complejidad que

involucra la creación de una arquitectura, se optó por usar una arquitectura pre-entrenada. En concreto, se decidió usar el modelo de redes residuales de 18 capas (K. He, Zhang, Ren, & Sun, 2016). Asimismo, al igual que con los data sets anteriores se implementó el modelo clásico y modelo con PCA. Sin embargo, para el caso de este data set, se optó por usar diferentes estrategias para la construcción de los datos con PCA. Es así que, por cada una de estas estrategias se obtuvieron diferentes modelos PCA. Estos modelos difieren principalmente del tipo de entrada utilizado. La Tabla 9 muestra una descripción de las dimensiones y tipo de características de cada arquitectura.

*Tabla 9: Descripción de modelos para data set Imágenes-3C.*

| <b>Modelo</b>              | <b>Dimensiones</b> | <b>Características</b> |                      |                      |
|----------------------------|--------------------|------------------------|----------------------|----------------------|
|                            |                    | <b>C<sub>1</sub></b>   | <b>C<sub>2</sub></b> | <b>C<sub>3</sub></b> |
| Resnet18-normal            | 224 x 224 x 3      | R                      | G                    | B                    |
| Resnet18-PCA-weak          | 224 x 224 x 3      | W                      | W                    | W                    |
| Resnet18-PCA-strong        | 224 x 224 x 3      | S                      | S                    | S                    |
| Resnet18-PCA-combined      | 224 x 224 x 3      | S                      | W                    | R <sub>3</sub>       |
| Resnet18-PCA-reconstructed | 224 x 224 x 3      | R <sub>1</sub>         | R <sub>2</sub>       | R <sub>3</sub>       |

Fuente: Elaboración propia. Modelos Resnet pre-entrenados

Como se puede apreciar, las entradas de los modelos PCA, representan diferentes combinaciones de las características obtenidas con PCA. Como se está trabajando con imágenes de tres canales, es posible usar las características de esta manera. En ese sentido, se consideró las características con menor (*weak*) y mayor (*strong*) varianza, así como también las características reconstruidas (*reconstructed*). Asimismo, el método clásico está representado por el modelo que usa las imágenes con sus canales RGB originales. Finalmente, la arquitectura usada por

todos los modelos se muestra en la Figura 6.

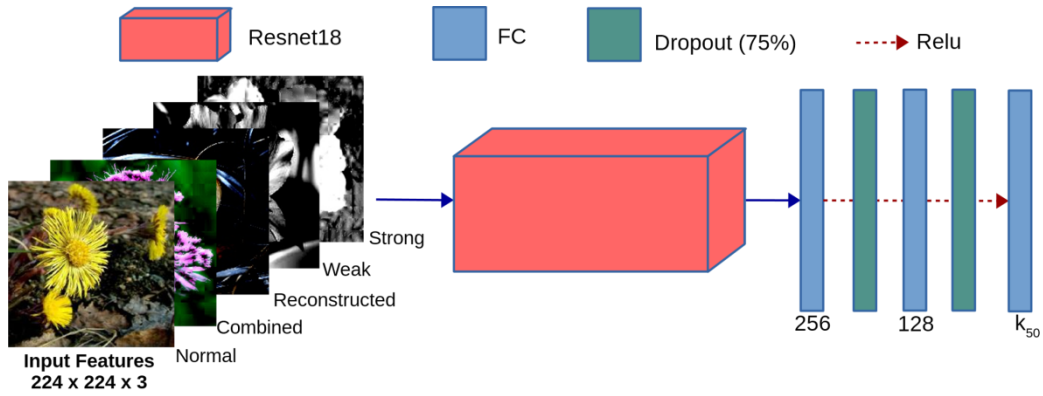


Figura 6: Arquitectura de modelos en data set Imágenes-3C.

Cabe resaltar que, tomando en cuenta todos los modelos (PCA y método clásico), a lo largo de todos los data sets, se entrenó un total de 38 modelos.

### 3.7. ANALISIS DE RESULTADOS

Esta etapa tiene por objetivo analizar los resultados obtenidos de los modelos entrenados. Para poder evaluar la performance de los modelos se toma como métrica a *accuracy* (exactitud), la cual se define como:

$$accuracy(y, \hat{y}) = \frac{1}{m} \sum_{i=0}^m 1. (\hat{y}_i = y_i)$$

Finalmente, en la Figura 7 se muestra una vista compacta de todos los pasos de la metodología propuesta en esta sección.

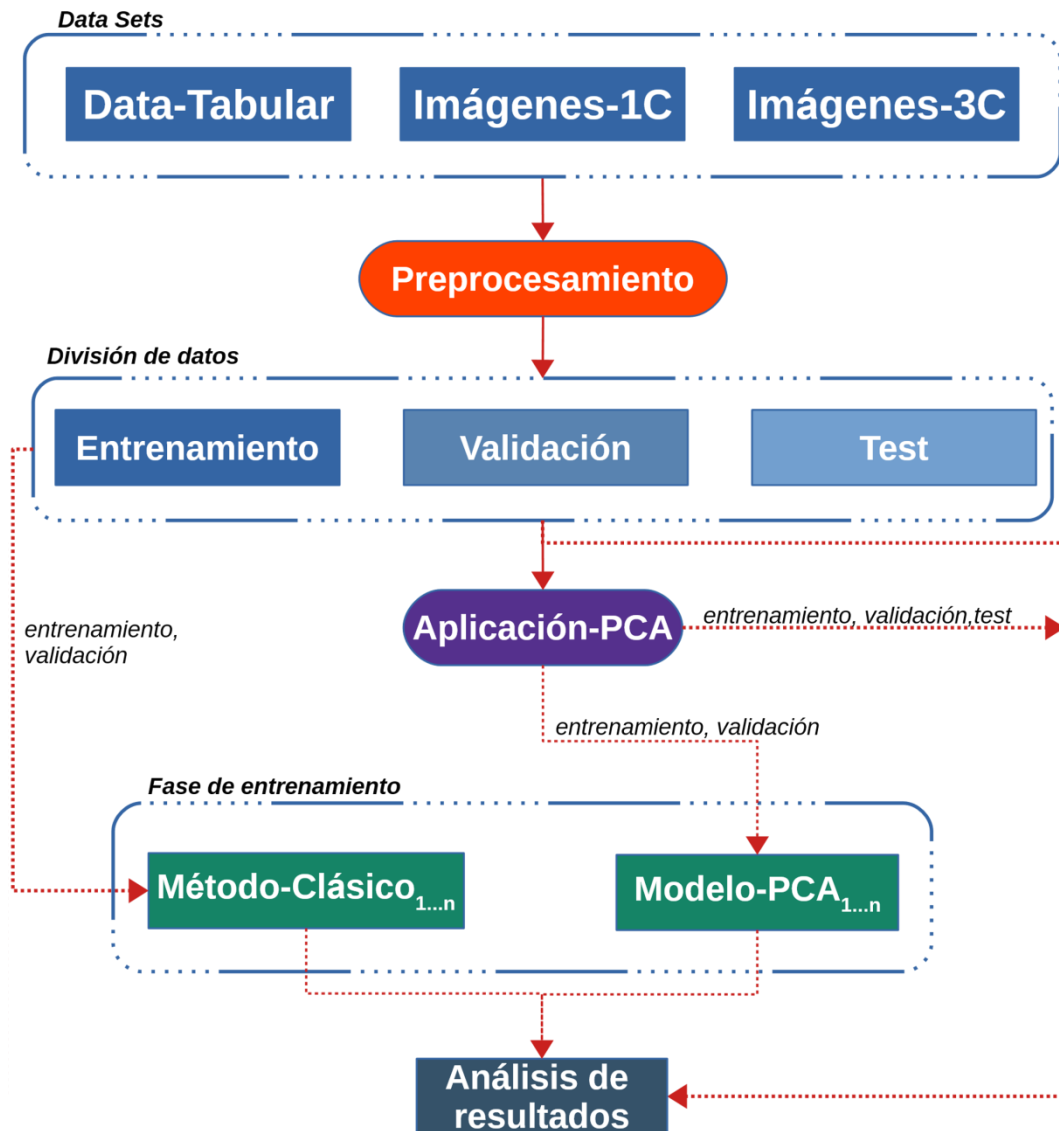


Figura 7: Vista global de la metodología.

### 3.8. TIPO DE INVESTIGACIÓN

El tipo de investigación es experimental. Según (Behar, 2008) este tipo de investigación también recibe el nombre de práctica, activa, dinámica. Se caracteriza porque busca la aplicación o utilización de los conocimientos que se adquieren. La investigación aplicada se encuentra estrechamente vinculada con la investigación básica, pues depende de los resultados y avances de esta última; esto queda aclarado si nos percatamos de que toda investigación aplicada requiere de un marco

teórico. Busca confrontar la teoría con la realidad. Es el estudio y aplicación de la investigación a problemas concretos, en circunstancias y características concretas. Esta forma de investigación se dirige a su aplicación inmediata y no al desarrollo de teorías.

La investigación aplicada, movida por el espíritu de la investigación fundamental, ha enfocado la atención sobre la solución de teorías. Conciérne a un grupo particular más bien que a todos en general. Se refiere a resultados inmediatos y se halla interesada en el perfeccionamiento de los individuos implicados en el proceso de la investigación. Sin embargo, en una investigación empírica, lo que le interesa al investigador, primordialmente, son las consecuencias prácticas. Si una investigación involucra problemas tanto teóricos como prácticos, recibe el nombre de mixta.

En realidad, un gran número de investigaciones participa de la naturaleza de las investigaciones básicas y de las aplicadas. Por la clase de medios utilizados para obtener los datos: documental, de campo o experimental. **Investigación documental:** Este tipo de investigación es la que se realiza, como su nombre lo indica, apoyándose en fuentes de carácter documental, esto es, en documentos de cualquier especie. Como subtipos de esta investigación encontramos la investigación bibliográfica, la hemerográfica y la archivística; la primera se basa en la consulta de libros, la segunda en Introducción a la metodología de la investigación, artículos o ensayos de revistas y periódicos y la tercera en documentos que se encuentran en los archivos, como cartas, oficios, circulares, expedientes, etcétera.

### **3.9. POBLACIÓN Y MUESTRA**

Para el caso de esta investigación, la población viene representada por los diferentes tipos de data sets definidos en la sección 3.2. Asimismo, en la sección 3.4 se muestra los porcentajes usados para cada tipo de data set. Por tanto, la población está representada por la totalidad de elementos de los diferentes conjuntos, el cual está dividido en datos de entrenamiento, validación y test.

### **3.10. TÉCNICAS Y HERRAMIENTAS**

Para la implementación de la metodología y el desarrollo en general de esta tesis se empleó las librerías PyTorch (Paszke et al., 2017), la cual dio soporte para la creación y validación de los modelos. Asimismo, para las demás tareas de la metodología se utilizó scikit-learn (Pedregosa et al., 2011). También se ha usado el organizador ADAM, las redes neuronales convolucional para el entrenamiento de set 3C y la ResNet para la transferencia de aprendizaje. Para optimizar los errores de los pesos se ha usado los algoritmos Backpropagation y el algoritmo de gradiente descendente.

## CAPÍTULO IV: RESULTADOS DE LA INVESTIGACIÓN

### 4.1. DATA SET TABULARES

Una vez finalizado el entrenamiento, se pudo observar que la aplicación de PCA permitió una reducción considerable en las dimensiones de los data sets tabulares. Al mismo tiempo, se logró retener una varianza igual o mayor al 90%. En este sentido, es importante notar que no siempre la varianza se mantenía en este margen. Esto se debe principalmente a que el número de componentes principales que mantienen la cantidad de varianza requerida en la fase de aplicación de PCA no siempre es exacto. En la Tabla 10 se detalla el cambio en las dimensiones de los data sets, así como también el porcentaje de varianza retenido.

Tabla 10: Efecto de PCA sobre los datos tabulares.

| DATA SET   | DIMENSIONES |     |         |
|--|-------------|-----|---------|
|  | Original    | PCA | Var (%) |
| GAMETES_Epistasis_2-Way_1000atts_0.4H_EDM-1_EDM-1_1        | 1000        | 575 | 90.00   |
| agaricus-lepiota   | 22          | 13  | 91.00   |
| GAMETES_Epistasis_2-Way_20atts_0.1H_EDM-1_1                | 20          | 18  | 92.00   |
| GAMETES_Epistasis_2-Way_20atts_0.4H_EDM-1_1                | 20          | 18  | 92.00   |
| GAMETES_Epistasis_3-Way_20atts_0.2H_EDM-1_1                | 20          | 18  | 92.00   |
| GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_50_EDM-2_001 | 20          | 18  | 92.00   |
| GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_75_EDM-2_001 | 20          | 18  | 92.00   |
| buggyCrx   | 15          | 12  | 94.00   |
| australian   | 14          | 11  | 90.00   |
| analcata_data_fraud  | 11          | 8   | 90.00   |
| analcata_data_cyyoung8092                                  | 10          | 5   | 92.00   |
| analcata_data_japansolvent                                 | 9           | 5   | 94.00   |
| analcata_data_creditscore                                  | 6           | 5   | 93.00   |
| analcata_data_aids   | 4           | 3   | 94.00   |
| analcata_data_lawsuit                                      | 4           | 3   | 92.00   |

Fuente: Elaboración propia. Transformación de dimensiones

Por otro lado, PCA puede aplicarse para la visualización de datos altamente dimensionales. En este caso, todos los data sets usados en esta sección presentan una dimensionalidad mayor a 3 dimensiones. Por ello,

realizar una visualización usando técnicas tradicionales resulta complicado.

Este escenario se acentúa aún más con el primer data set de la Tabla 10, el cual posee 1000 dimensiones. Sin embargo, usando PCA, es posible realizar dichas visualizaciones si se proyectan algunos de los componentes principales con mayor varianza. La Figura 8 muestra la visualización de los data sets tabulares de acuerdo al orden descrito en la Tabla 10, usando como base los dos primeros componentes principales de cada data set.

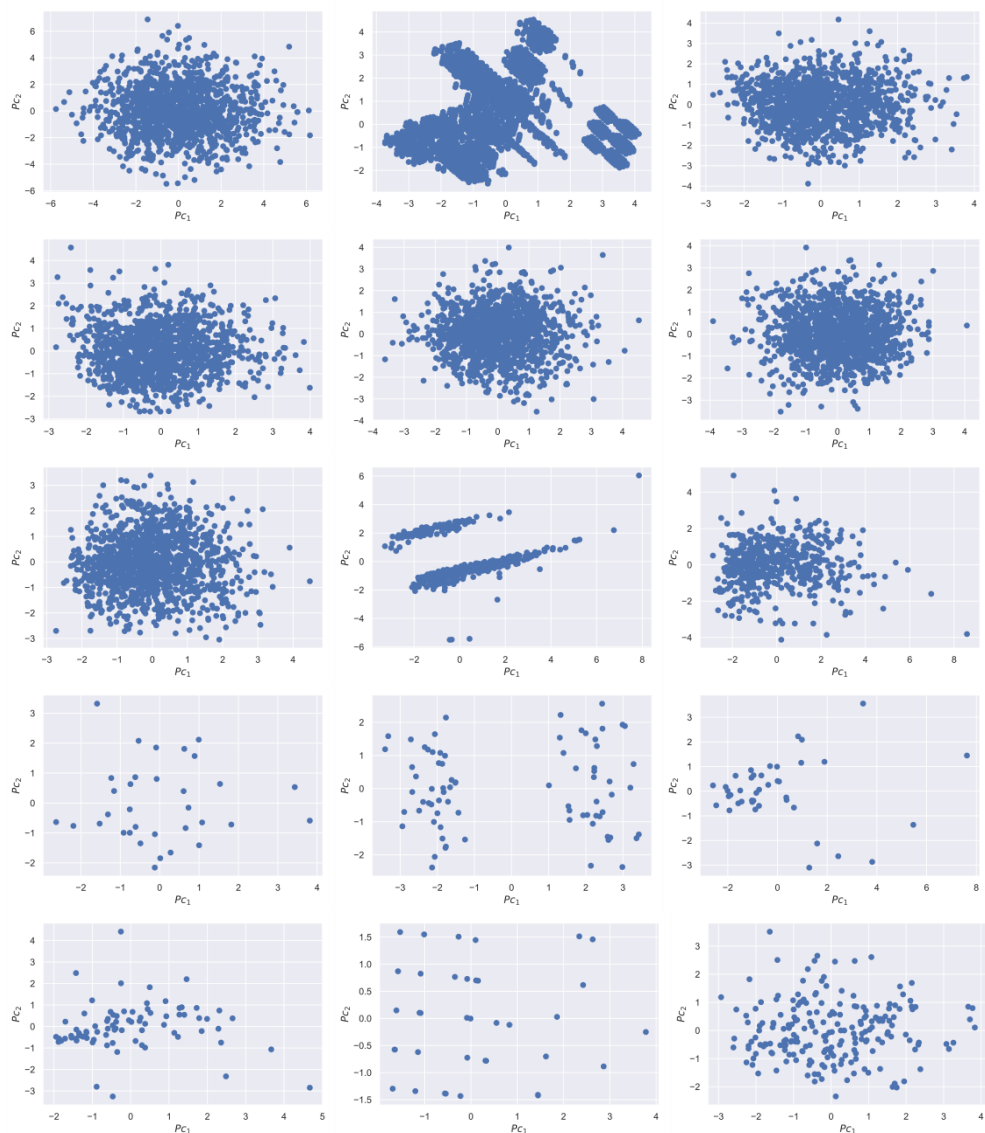


Figura 8: Visualización de componentes principales para los data set tabulares.



Como se puede observar, en los nueve primeros data sets, se logra centrar los datos. Esto significa que los dos primeros componentes principales que se han seleccionado logran mantener la mayoría de información para la visualización. Asimismo, esta visualización nos brinda una descripción de la distribución de los datos. Para el caso de los nueve primeros data sets, nos permite observar ciertas tendencias en los dos componentes principales. Sin embargo, no siempre se obtiene una visualización adecuada, y es que, es necesario realizar diversas combinaciones hasta encontrar una representación informativa. Esta situación se corrobora con el resto de data sets. Donde juntando los dos componentes principales no es posible observar una distribución de los datos. Sin embargo, esto puede simplemente sugerir que otros componentes son más adecuados para la visualización. Respecto a la performance de aprendizaje, en la Tabla 11 se muestra una comparación al usar el método clásico y PCA. Esta comparación se extiende por los tres grupos de datos: entrenamiento, validación y test para cada uno de los data sets tabulares. En particular se presta mayor atención a los valores obtenidos en los datos de test. Ya que, miden perfectamente la capacidad de generalización de los modelos. Como se puede observar, existe una diferencia en los resultados obtenidos para los tres grupos. En particular, se nota que, para algunos data sets, tanto el método clásico como PCA logran obtener una mejor performance en los datos de test. Asimismo, se observan ciertos problemas de *overffiting* para ambos modelos con respecto a los datos de entrenamiento. Esto se debe principalmente a que, al usar una arquitectura para todos los data sets, es posible que esta sea demasiado compleja (*overffiting*). Asimismo, se observa que para el caso de los data sets: agaricus-lepiota y analcatdata\_aids, ambos modelos

obtienen la misma performance en los datos de test. Exploraremos este y otros fenómenos a más profundidad en la sección de CONCLUSIONES.

Tabla 11: Comparación de performance de aprendizaje entre PCA y el Método Clásico en data set tabulares.

| DATA SET   | ACCURACY (%)  |            |               |                |            |               |
|--|---------------|------------|---------------|----------------|------------|---------------|
|  | PCA           |            |               | MÉTODO CLÁSICO |            |               |
|  | Entrenamiento | Validación | Test          | Entrenamiento  | Validación | Test          |
| GAMETES_Epistasis_2-Way_1000atts_0.4H_EDM-1_EDM-1_1        | 78.44         | 49.38      | <b>45.00</b>  | 85.63          | 55.00      | 43.13         |
| agaricus-lepiota   | 100.00        | 100.00     | 100.00        | 100.00         | 100.00     | 100.00        |
| GAMETES_Epistasis_2-Way_20atts_0.1H_EDM-1_1                | 74.69         | 59.38      | 50.63         | 73.05          | 58.75      | <b>51.88</b>  |
| GAMETES_Epistasis_2-Way_20atts_0.4H_EDM-1_1                | 79.30         | 61.88      | 62.50         | 80.78          | 67.50      | <b>65.00</b>  |
| GAMETES_Epistasis_3-Way_20atts_0.2H_EDM-1_1                | 57.58         | 56.25      | <b>54.38</b>  | 56.88          | 48.13      | 49.38         |
| GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_50_EDM-2_001 | 79.92         | 66.88      | <b>63.13</b>  | 80.94          | 71.88      | 61.88         |
| GAMETES_Heterogeneity_20atts_1600_Het_0.4_0.2_75_EDM-2_001 | 79.06         | 63.75      | <b>71.25</b>  | 81.17          | 67.50      | 67.50         |
| buggyCrx   | 86.59         | 84.10      | <b>92.75</b>  | 87.32          | 86.96      | 91.30         |
| australian   | 88.77         | 78.26      | <b>84.06</b>  | 92.75          | 82.61      | 82.61         |
| analcata_data_fraud  | 93.94         | 75.00      | 40.00         | 93.94          | 75.00      | <b>60.00</b>  |
| analcata_data_cyyoung8092                                  | 81.82         | 80.00      | <b>80.00</b>  | 83.12          | 80.00      | 70.00         |
| analcata_data_japansolvent                                 | 90.24         | 80.00      | 83.33         | 100.00         | 80.00      | <b>100.00</b> |
| analcata_data_creditscore                                  | 86.25         | 80.00      | 80.00         | 82.50          | 70.00      | <b>90.00</b>  |
| analcata_data_aids   | 45.00         | 60.00      | <b>100.00</b> | 62.50          | 20.00      | 40.00         |
| analcata_data_lawsuit                                      | 98.10         | 100.00     | 96.30         | 99.53          | 100.00     | 96.30         |
| <b>AVG</b>   | 81.31         | 72.99      | <b>73.56</b>  | 84.01          | 70.89      | 71.27         |

Fuente: Elaboración propia. Efecto del PCA sobre los datos tabulares

#### 4.2. DATA SET DE IMÁGENES – 1C

Los resultados obtenidos al aplicar PCA a las imágenes de un solo canal muestran otra vez el potencial de PCA para la reducción de dimensiones con una pérdida aceptable. Como se observa en la Tabla 12, el data set de imágenes de un solo canal obtuvo una reducción significativa de sus dimensiones, pasando de 784 a 144. Del mismo modo, se logró retener una varianza del 90%.

*Tabla 12: Efecto de PCA sobre imágenes-1C.*

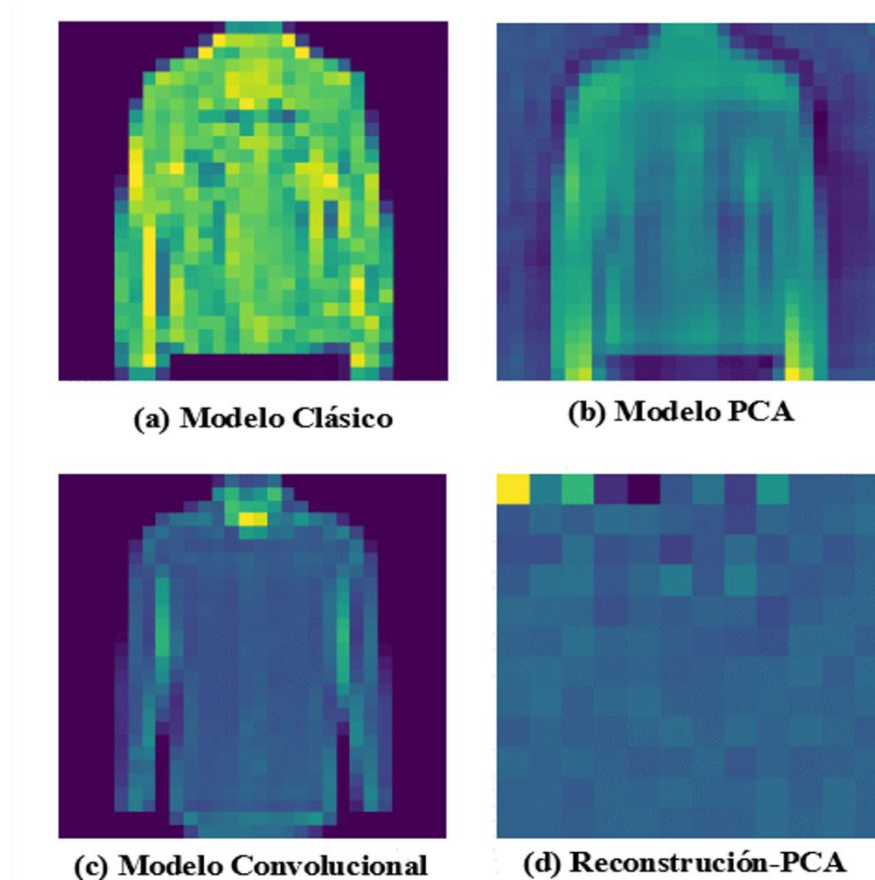
| DATA SET      | DIMENSIONES |     |         |
|---------------|-------------|-----|---------|
|               | Original    | PCA | Var (%) |
| Fashion-MNIST | 784         | 144 | 90.00   |

Fuente: Elaboración propia

Respecto a las visualizaciones, es importante notar las características de cada representación. Como se puede apreciar en la Figura 9, el modelo clásico (a) hace uso de los datos con sus dimensiones originales, esto representa un vector de 784 dimensiones. Por el contrario, el modelo PCA (b) usa solamente 144 dimensiones. Computacionalmente, esto se traduce en una representación menos costosa en términos de almacenamiento (RAM, HDD, etc.). Sin embargo, como veremos más adelante, este hecho no siempre es cierto, especialmente para el caso de imágenes de mayores dimensiones, donde es necesario utilizar otras técnicas.

Ahora, el modelo convolucional (c) tiene una ventaja en comparación con los otros modelos. Ya que, no necesita utilizar una representación vectorial. Por el contrario, recibe la imagen con sus dimensiones originales: 28 x 28. Este hecho podría parecer trivial, sin embargo, si se tuviese una imagen de 250 x 250 x 3, el vector resultante sería  $1875 \times 10^2$ , ahora apliquemos este cálculo para 5K imágenes y tendríamos una matriz de

5K x 250 x 250 x 3, lo que resultaría en  $9375 \times 10^5$  datos para almacenar. Otro punto a favor del modelo convolucional (c) es la capacidad de usar filtros para la detección de características, ventaja no presente en sus contrapartes (a) y (b). Ahora, bien, dada la naturaleza de PCA es posible reconstruir la imagen original a partir de la representación de componentes principales. Esto se observa en la imagen (d), sin embargo, a diferencia de las otras representaciones, no se puede apreciar una reconstrucción aceptable de la clase original. Esto se debe al número de componentes principales (144). Es posible que, si se incrementan los componentes, la reconstrucción sea más fiel al original.



*Figura 9: Visualización de datos de modelo clásico (a), PCA (b), convolucional (c) y reconstrucción de PCA (d).*

Respecto a la performance de aprendizaje, en la Tabla 13 se muestra la comparación respectiva entre los diferentes modelos.

Tabla 13: Comparación de performance de aprendizaje entre PCA y el Método Clásico en imágenes-1C.

| Modelos                | Data set      |            |                | M-Epochs |
|------------------------|---------------|------------|----------------|----------|
|                        | Entrenamiento | Validación | Test           |          |
| Modelo Clásico         | 93.74 %       | 89.03 %    | 88.49 %        | 58       |
| Modelo PCA             | 97.71 %       | 89.16 %    | <b>88.80 %</b> | 45       |
| Modelo Convolutacional | 90.24 %       | 88.68 %    | 88.55 %        | 50       |

Fuente: Elaboración propia. Accuracy de las divisiones de la data set

Como se puede observar, para los datos de test, el modelo con PCA logra obtener la mayor performance de aprendizaje. Asimismo, de las 100 iteraciones, el modelo con PCA obtiene su mayor performance en la 45ava iteración. Esto nos muestra una ventaja significativa con respecto a los beneficios que tiene PCA sobre los tiempos de entrenamiento (*epochs*). Sin embargo, el modelo convolutacional obtiene una performance similar al modelo con PCA. De hecho, la diferencia es pequeña: **0.25**. Finalmente, el modelo clásico obtiene la menor performance, a la vez que su mejor iteración es más alta en comparación con los otros modelos. Como se puede notar, casi todos los modelos obtienen una performance alta. Esto se debe principalmente a la potencia que tienen las redes neuronales para aprender patrones. No obstante, como veremos, al aplicar la misma técnica con imágenes en 3 colores, los resultados ya no serán tan parecidos. Finalmente, se logra observar que el modelo con PCA presenta *overffiting* en los datos de entrenamiento.

#### 4.3. DATA SET DE IMÁGENES - 3C

De todos los data sets, los de imágenes en tres canales son los más complejos. La complejidad de este data set se deriva en: (1) número de

clase, (2) distribución de clases, (3) variaciones en luz, tamaño, contrastes, etc., y (4) dimensiones. Debido a este último factor (4), se tuvo que aplicar una variante en línea de PCA (Ross, Lim, Lin, & Yang, 2008), tomando lotes de 500 imágenes a la vez. Asimismo, la aplicación de PCA se orientó a los canales RGB en lugar de a las dimensiones como se hizo con el data set de imágenes-1C. Como se muestra en la Tabla 14, el data set de Imágenes en tres canales, obtiene una reducción de una sola dimensión. Asimismo, la varianza retenida es del 91%.

*Tabla 14: Efecto de PCA sobre imágenes-3C.*

| DATA SET | DIMENSIONES |     |         |
|----------|-------------|-----|---------|
|          | Original    | PCA | Var (%) |
| Flower   | 3           | 2   | 91.00   |

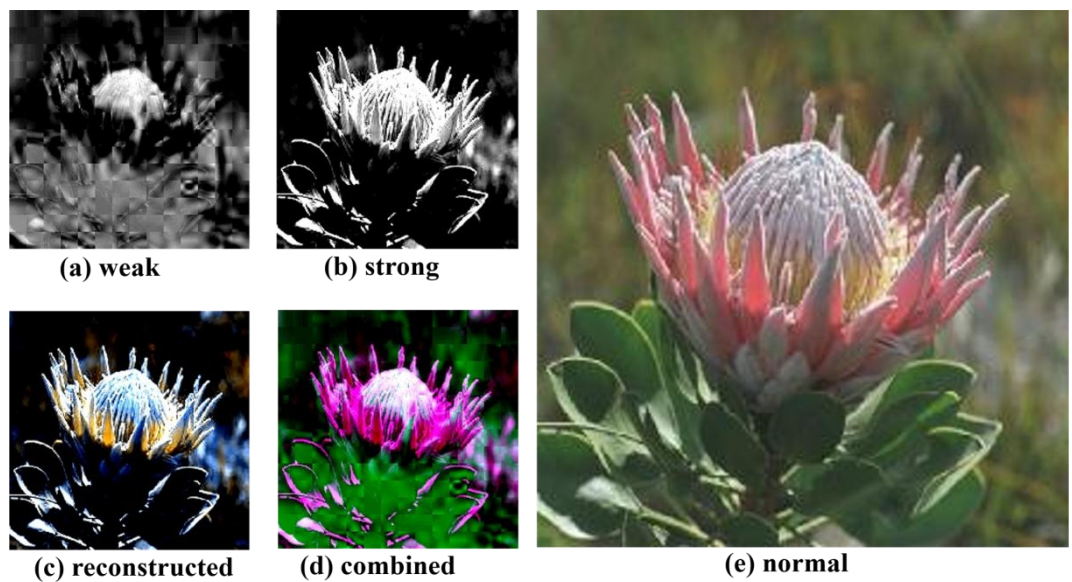
Fuente: Elaboración propia

Ahora, respecto a las visualizaciones, como se muestra en la Figura 10 cada representación obtenida al combinar las características de PCA se caracteriza por una porción de la imagen que resaltan y una que pierden.

Por ejemplo, la representación débil (a) mantiene resalta algunos contornos de la imagen, sin embargo, no es capaz de mantener gran parte de las características de la imagen original (e). Esto sucede porque este componente solo mantiene la varianza más baja. De hecho, sino se tuviera el contexto, no se podría inferir que se trata de una flor. Por el contrario, la representación fuerte (b) mantiene los bordes de la imagen, así como también otros detalles de la flor, sin embargo, pierde gran parte del contenido del fondo. No obstante, ahora es posible distinguir la flor.

Continuando con la representación reconstruida (c), esta implementa cierta tonalidad sobre el centro de la flor, asimismo reconstruye parcialmente el fondo de la imagen. No obstante, esta representación no

es capaz de reconstruir la imagen (e) a su totalidad. Observando esta imagen podría pensarse que se aumentó deliberadamente el contraste. Esto se debe a que, se ha perdido el 9% de la varianza, por ende, la reconstrucción no es perfecta. Finalmente, la representación combinada (d) agrega parte de los colores originales de la imagen, así como cierta porción del fondo. Sin embargo, no es capaz de reconstruir adecuadamente la parte verde inferior de la flor original (e). Indistintamente, cada representación acentúa una característica específica, ya sea: bordes, fondo, colores, formas, etc. A la vez que pierde una porción de la imagen original (e).



*Figura 10: Visualización de características usando PCA (a) - (d) y datos normales (e).*

Esta falta de reconstrucción y variación en las representaciones se plasma directamente al comparar la performance de los modelos. Como se puede observar en la Tabla 15, todos los modelos con PCA, obtienen una performance inferior en comparación con el modelo que uso los datos originales. Asimismo, casi todos los modelos lograron obtener su máxima



performance cerca a las 100 iteraciones. En la sección de CONCLUSIONES ahondaremos más en los detalles de este fenómeno.

Tabla 15: Comparación de accuracy entre modelos pre-entrenados

| Modelos                    | Data set      |            |       | M-Epochs |
|----------------------------|---------------|------------|-------|----------|
|                            | Entrenamiento | Validación | Test  |          |
| Resnet18-Normal            | 93.03         | 91.40      | 86.16 | 98       |
| Resnet18-PCA-weak          | 54.32         | 47.41      | 46.59 | 97       |
| Resnet18-PCA-strong        | 76.15         | 69.24      | 66.86 | 99       |
| Resnet18-PCA-combined      | 80.31         | 73.32      | 71.35 | 100      |
| Resnet18-PCA-reconstructed | 84.33         | 78.72      | 76.02 | 95       |

Fuente: Elaboración propia. Accuracy

#### 4.4. VALIDACIÓN DE HIPÓTESIS

##### Hipótesis General (HG)

**Existe una diferencia significativa en la implementación de la red neuronal *backpropagation* usando los resultados del PCA y del método clásico.**

Esta diferencia se acentúa fuertemente para los datos tabulares, donde el uso de PCA logró incrementar la performance de aprendizaje (*accuracy*) para los datos de test, con un promedio del **73.56%** en comparación con el método clásico: **71.27%**. Asimismo, el uso de PCA tuvo una mejora en *accuracy* para los datos de imágenes de un solo canal. Sin embargo, esta mejora no es tan alta como con los datos tabulares. De hecho, de los tres modelos usados (método clásico, método PCA y red convolucional), el método con PCA fue superior en un **0.31%** y **0.25%** en comparación con el método clásico y red convolucional respectivamente. Asimismo, usando PCA, el número de iteraciones fue menor (45) en comparación con el método clásico (58) y convolucional (50). Asimismo, al observar los

resultados en accuracy de los datos de imágenes de tres canales, no se obtuvo mejoras significativas al usar PCA. De hecho, de las cuatro estrategias de PCA aplicadas, el resultado para los datos de test se mantuvo inferior respecto al **86.16%** de *accuracy* obtenido por el método clásico. En ese sentido, los modelos con PCA, obtuvieron un: **46.59%**, **66.86%**, **71.35%** y **76.02%** para los modelos: *weak*, *strong*, *combined* y *reconstructed* respectivamente.

Asimismo, se observó una tendencia hacia *overfitting* por parte de todos los modelos. Esto se debe a que se usó una arquitectura constante por cada tipo de data set, manteniendo la misma estructura de hiperparámetros.

#### **Hipótesis Específicas (HE)**

- **HE1:** Aplicando el PCA en el tratamiento de datos multivariantes se reduce la dimensionalidad de las variables.

Este hecho se pudo corroborar con los tres tipos de data sets. Siendo el caso de los datos tabulares los más prominentes. Asimismo, para los datos de imágenes de un solo canal, el uso de PCA, logro disminuir el tamaño de las imágenes. Esta reducción de dimensiones se mantuvo conjuntamente con una consistencia en los datos. Esto se comprueba al observar los resultados de *accuracy* de los modelos para los dos primeros tipos de data sets. Asimismo, para los datos de imágenes de tres canales, se logra también reducir las dimensiones. Sin embargo, esta reducción no resultó en una mejor significativa para la clasificación.

- **HE2:** Los resultados del PCA influyen en la implementación de la RNA. Esto se pudo corroborar al momento de implementar las diversas arquitecturas para los diferentes tipos de data sets. Ya que, si bien es

cierto, todas ellas compartían una misma arquitectura, el efecto de PCA se manifestó en el tamaño del vector de entrada para los diferentes modelos.

- **HE3:** PCA, mejora la representación geométrica de los datos obtenidos mediante el uso de un *framework*.

Esto es particularmente cierto, ya que, la implementación en código desde cero de PCA, implica la optimización de diversas operaciones matemáticas, las cuales ya se encuentran optimizadas en los *frameworks* usados en esta investigación.

- **HE4:** Los resultados del PCA influyen en el entrenamiento de la RNA.

Como se ha visto en la sección de resultados. PCA influye tanto en la performance de aprendizaje obtenida por los modelos, como en el tiempo de entrenamiento. Esto se observó para los datos tabulares y las imágenes de un solo canal. Donde, gracias a la reducción de dimensiones se pudo obtener una performance promedio del: **73.56%** en comparación con **71.27%** del método clásico para los datos tabulares. Mientras que para los datos de imágenes de un solo canal se obtuvo una mejora del **0.31%** y **0.25%** en comparación con el método clásico y red convolucional sobre una base del **88.00%** de *accuracy* para los tres modelos. Asimismo, PCA también tuvo una influencia en performance para los datos de imágenes de tres canales. Sin embargo, a diferencia de los dos data sets anteriores, este efecto se tradujo en una disminución considerable en *accuracy* del: **46.59%**, **66.86%**, **71.35%** y **76.02%** para los modelos: *weak*, *strong*, *combined* y *reconstructed* en comparación con el **86.16%** obtenido por el método clásico.

- **HE5:** La métrica definida sobre el espacio data set garantiza la convergencia.

El espacio  $E = \{x = (x_1, x_2, \dots, x_n) / x_i \in \mathbb{R}\}$  es llamado Espacio Euclidiano, que es espacio Vectorial, Producto Interno, Normado, Métrico, Completo y de Hilbert. Pero también espacio de Banach.

El espacio  $M^{m \times n} = \{A = [a_{ij}]_{m \times n} \mid a_{ij} \in \mathbb{R}\}$ , representa al espacio de las matrices con “m” filas y “n” columnas con elementos reales, que podemos representarlo como  $M$ .

Luego  $\langle M^{m \times n}, +, *, \mathbb{R} \rangle$ , representa el espacio vectorial de matrices sobre el campo  $\mathbb{R}$ , con la operación de suma de matrices y el producto por un escalar. Sobre este espacio definimos el producto interno de la siguiente manera:

$$\langle, \rangle: M \times M \rightarrow \mathbb{R} / \langle A, B \rangle = \text{Tr}(B^t A) \text{ y como } \langle x, x \rangle = \|x\|^2,$$

$M$  se transforma en espacio normado primero y luego en espacio métrico. Una de las formas es definir:

$$d(A, B) = \|A - B\| = \sup_{x \in \langle 0, 1 \rangle} \left\{ \frac{\|(A - B)x\|}{\|x\|} \mid \|x\| = 1 \right\}$$

Al considerar los espacios data set como matrices de números reales, asimismo la matriz de covarianza, tiene similar estructura, pueden considerarse subespacios de la matriz  $M$ , en estos espacios se definen en algunos casos métricas euclidianas y productos internos tanto de vectores como de matrices lo que transforma  $M$  en un espacio producto interno (Conway, 1990). Finalmente, la convergencia implica la existencia de sucesiones que convergen a números reales.

## CONCLUSIONES

- Para los data sets tabular se observó una mejora respecto a la disminución de dimensiones y aumento de performance de aprendizaje. Ya que, de los 15 data sets, **8 (8/15) 53.33%** obtuvieron una mejora en los datos de test usando PCA en comparación con el **(5/15) 33.33%** usando el método clásico. Asimismo, ambos modelos obtuvieron resultados iguales en un total de **(2/15) 13.33%**. Este hecho también se refleja al observar el promedio de los resultados para los datos de test, donde PCA obtiene un total de **73.56%** en comparación con el método clásico: **71.27%**. No obstante, también se observaron problemas de *overffiting* para la gran mayoría de modelos (método clásico y PCA). La razón reside en la arquitectura usada, ya que, al usar una misma arquitectura, esta podría resultar muy compleja (*overffiting*) para el tipo de data set.
- Al ser el PCA, es una técnica útil al momento de visualizar datos que se encuentran en espacios altamente dimensionales. En particular, para los datos tabulares, fue posible crear una visualización en 2D. No obstante, este método requiere probar diferentes combinaciones de componentes principales para una correcta interpretación. Asimismo, desde el punto de vista del entrenamiento, el aplicar PCA en los datos tabulares disminuye el tiempo de entrenamiento, ya que se trabaja con menos dimensiones.
- Al igual que con los datos tabulares, las imágenes de un solo canal se beneficiaron de la reducción de dimensiones y el aumento en performance de aprendizaje. No obstante, ambos modelos presentarían problemas si las

dimensiones de las imágenes aumentarían. En ese sentido, las redes convolucionales presentan una gran ventaja en comparación con los modelos totalmente conectados. Por otro lado, los resultados en la performance de aprendizaje de los modelos, presentan variaciones pequeñas en los datos de test. Claramente el modelo con PCA supera a los modelos clásico y convolucional. No obstante, este modelo es el que más *overfitting* presenta. Esto podría solucionarse si se agregarían ciertas medidas de restricciones al modelo, como cambio de pesos o modificación de la ratio de aprendizaje.

- Para los data set de imágenes en tres canales, no se pudo obtener una performance de aprendizaje igual o superior al modelo clásico, inclusive realizando diversas combinaciones. La razón fundamental para este resultado reside en la naturaleza de las redes convolucionales. Ya que, cada capa extrae características, al ser procesadas por PCA, estas ingresan con la pérdida de información importante. Por ende, los modelos no son capaces de generar una representación adecuada de los datos para su clasificación. En otras palabras, en el caso de las redes convolucionales, y más aún en modelos pre-entrenados, PCA no mostro mejora alguna en comparación con el modelo clásico. No obstante, es posible obtener beneficios de PCA al usarse dentro de la red, como una capa, ya que esta disminución de características resultaría útil, sin embargo, el efecto no es el mismo cuando se usa directamente como datos de entrada.

## RECOMENDACIONES

En la realización de este trabajo se sugieren las siguientes recomendaciones:

- Agregar técnicas de regularización para todos los modelos para prevenir el *overfitting*.
- Utilizar diferentes optimizadores para las etapas de entrenamiento, de ser posible usar un optimizador diferente para cada data set.
- Recolectar más datos para el entrenamiento de modelos pre-entrenados
- Aplicar otras técnicas de preprocesamiento por cada data set.

## REFERENCIAS BIBLIOGRÁFICAS

- ASonawale, S., & Ade, R. (2015). Dimensionality Reduction: An Effective Technique for Feature Selection. *International Journal of Computer Applications*, 117(3), 18–23. <https://doi.org/10.5120/20535-2893>
- Awate, G., Bangare, S., Pradeepini, G., & Patil, S. (2018). Detection of Alzheimers Disease from MRI using Convolutional Neural Network with Tensorflow, 1–15. <https://doi.org/https://arxiv.org/abs/1806.10170>
- Behar, D. (2008). *Metodología de la Investigación*. (Shalom, Ed.).
- Bravo, I. (2007). *Arquitectura basada en FPGAs para la detección de objetos en movimiento , utilizando visión computacional y técnicas*. Universidad de Alcalá.
- Castrillón, O. & Rodríguez, M. & Leyton, L. (2008). ÉTICA E INTELIGENCIA ARTIFICIAL ¿ NECESIDAD O URGENCIA ?
- Conitzer, V. (2016). PHILOSOPHY IN THE FACE OF ARTIFICIAL. *Cs.AI*, 1–6.
- Conway, J. B. (1990). *A course in Functional analysis*. Springer-Verlag. <https://doi.org/10.1017/CBO9781107415324.004>
- Faik, M. (2016). *Face recognition in low-resolution images under small sample conditions with face-part detection aliganment*. University of Gromingen.
- Gurney, K. (2004). *Neural Networks* (British Li). London. [https://doi.org/10.1016/S0140-6736\(95\)91746-2](https://doi.org/10.1016/S0140-6736(95)91746-2)
- He, J., & Xu, J. (2019). MgNet: A Unified Framework of Multigrid and Convolutional Neural Network, 1–26.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). IEEE. <https://doi.org/10.1109/CVPR.2016.90>
- Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *AIP Conference Proceedings*, 1631, 58–62. Retrieved from <http://arxiv.org/abs/1412.6980>



- Kumar, N. V., Mathew, S., & Swaminathan, G. (2010). Analysis of Groundwater for Potability from Tiruchirappalli City Using Backpropagation ANN Model and GIS. *Journal of Environmental Protection*, 01(02), 136–142. <https://doi.org/10.4236/jep.2010.12018>
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40(2012), e253. <https://doi.org/10.1017/S0140525X16001837>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *NATURA International Weekly Journal of Science*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lippi, V., & Ceccarelli, G. (2019). Exact implementation and continuity corrections. *STAT.ML*.
- Nilsback, M.-E., & Zisserman, A. (2008). Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing* (pp. 722–729). IEEE. <https://doi.org/10.1109/ICVGIP.2008.47>
- Ojeda, E. (2016). *Reconocimiento de Patrones Basado en Redes Neuronales Artificiales*. Universidad Nacional de Piura.
- Olson, R. S., La Cava, W., Orzechowski, P., Urbanowicz, R. J., & Moore, J. H. (2017). PMLB: A large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(1). <https://doi.org/10.1186/s13040-017-0154-4>
- Palacios, M. E. B. (2014). *A PRENDIZAJE COMPUTACIONAL y MORFOLOGÍA MATEMÁTICA*. Universidad Nacional de Mar del Plata.
- Paszke, A., Chanan, G., Lin, Z., Gross, S., Yang, E., Antiga, L., & Devito, Z. (2017). Automatic differentiation in PyTorch. In *NIPS-W* (p. 4). Retrieved from <https://openreview.net/pdf?id=BJJsrnfCZ>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. Retrieved from

<http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>

Peña, A. (2016). *Uso de Redes Neuronales Artificiales para Optimizar la Dosificación de Coagulantes en la Planta de Tratamiento de Agua Potable en Huancayo*. Universidad Nacional Agraria La Molina.

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of adam and beyond. In *International Conference On Learning Representations*. Retrieved from <https://openreview.net/pdf?id=ryQu7f-RZ>

Ross, D. A., Lim, J., Lin, R.-S., & Yang, M.-H. (2008). Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, 77(1–3), 125–141. <https://doi.org/10.1007/s11263-007-0075-7>

Saenz, A & Martin, B. (2002). *Redes Neuronales y sistemas Difusos* (Segunda). Mexico D. F.

Sarveniazi, A. (2014). An Actual Survey of Dimensionality Reduction. *American Journal of Computational Mathematics*, 04(02), 55–72. <https://doi.org/10.4236/ajcm.2014.42006>

Vega, H. (2011). *Redes Neuronales para el Reconocimiento de la Calidad Morfológica de Mangos Exportables para la Empresa Bofruit Perú S.A.C*. Universidad Nacional de San Marcos.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Retrieved from <http://arxiv.org/abs/1708.07747>

Zhang, J. (2017). Multivariate Analysis and Machine Learning in Cerebral Palsy Research. *Frontiers in Neurology*, 8, 715. <https://doi.org/10.3389/fneur.2017.00715>

## **ANEXOS**

**ANEXO A: Matriz de consistencia**

**ANEXO B: Configuración de Hiper-parámetros**

## ANEXO A. MATRIZ DE CONSISTENCIA

| PROBLEMA   | OBJETIVOS   | HIPÓTESIS  | VARIABLES  | INDICADORES  | INSTRUMENTOS  |
|--|---|--|--|--|---|
| • Principal  | • General   | • General  | • Independientes   |  |   |
| ¿Es posible efectuar el análisis comparativo de los resultados obtenidos a partir de la implementación de la red neuronal <i>Backpropagation</i> mediante la aplicación del Análisis de Componentes Principales (PCA) y el método clásico? | Realizar el análisis comparativo en la implementación de la red neuronal <i>Backpropagation</i> usando el PCA y el esquema clásico. | Existe una diferencia significativa en la implementación de la red neuronal <i>Backpropagation</i> usando los resultados del PCA y del método clásico. | Análisis de Componentes Principales (PCA), método clásico.   | Número de Variables al final de la implementación del PCA.<br><br>Métrica de performance <i>Accuracy</i> | Lenguaje de programación Python.<br>Librería Pytorch.<br>Librería Scikit-learn.       |
| • Secundarios  | • Específicos   | • Específicas  | • Dependientes   |  |   |
| <b>PE1:</b> ¿En qué medida se reduce la dimensionalidad de las variables aplicando el PCA en el tratamiento de datos multivariantes?   | OE1: Reducir la dimensionalidad de las variables aplicando el PCA en el tratamiento de los datos multivariantes                     | HE1: Aplicando el PCA en el tratamiento de datos multivariantes se reduce la dimensionalidad de las variables.   | Análisis comparativo en la implementación de la red neuronal | Número de dimensiones.   | Librería Scikit-learn.  |
| <b>PE2:</b> ¿Cuál es el efecto de los resultados del PCA en la implementación de la RNA <i>Backpropagation</i> ?   | OE2: Determinar el efecto de los resultados del PCA en la Implementación de la red neuronal artificial (RNA) <i>Backpropagation</i> | HE2: Los resultados del PCA influyen en la implementación de la RNA.   |  | Número de vectores de entrada.   | Redes totalmente conectadas.<br>Redes convoluciones.<br>Redes preentrenadas resnet18. |
| <b>PE3:</b> ¿En qué medida PCA mejora la representación gráfica  | OE3: Determinar la mejora de la   | HE3: PCA, mejora la representación   |  | Número de componentes de   | Visualizaciones   |

|   |   |   |  |   |   |
|---|---|---|--|---|---|
| de los datos multivariantes mediante el uso de un framework?  | representación gráfica de los datos multivariantes Usando un framework adecuado           | geométrica de los datos obtenidos mediante el uso de un framework.  |  | principales.  |   |
| <b>PE4:</b> ¿En qué medida el uso de los resultados del PCA, influye en el entrenamiento de la RNA?   | OE4: Determinar la influencia en el entrenamiento de la RNA usando los resultados del PCA | HE4: Los resultados del PCA influyen en el entrenamiento de la RNA.   |  | Promedio de Métrica de performance<br><i>Accuracy</i><br><br><i>Número de iteraciones</i> | Redes totalmente conectadas.<br>Redes convoluciones.<br>Redes preentrenadas resnet18. |
| <b>PE5:</b> ¿En qué medida el hecho de que los espacios (Redes neuronales y espacio multivariable) sean métricos garantiza la convergencia? | OE5: Determinar si los espacios multivariantes son métricos                               | HE5: Si el espacio es métrico es posible la convergencia geométrica de los datos obtenidos mediante el PCA. |  | Convergencia.   | Sucesiones.   |

Fuente: Elaboración propia

## ANEXO B. CONFIGURACIÓN DE HIPER-PARÁMETROS

El entrenamiento de los modelos se realizó utilizando un acelerador GPU. La lista de configuraciones se aplicó para cada tipo de modelo, considerando la arquitectura y tipo de data set. Asimismo para el entrenamiento se usó ADAM (Kingma & Ba, 2014) conjuntamente con la variación AMSGRAD (Reddi, Kale, & Kumar, 2018) para una mejor estabilidad.

- **Data set Tabulares:**

*Tabla 16: Configuración para modelos en data set tabulares.*

| Hiper-parámetro      | Valor        |
|----------------------|--------------|
| Batch                | 64           |
| Ratio de aprendizaje | 5e-4         |
| Optimizador          | ADAM-AMSGRAD |
| Epochs               | 100          |
| Ratio dropout        | 30%          |

Fuente: Elaboración propia. Distribución de hiper- parámetros para datos tabulares

- **Data set Imágenes-1C:**

*Tabla 17: Configuración para modelos en data set Imágenes-1C.*

| Hiper-parámetro      | Valor        |
|----------------------|--------------|
| Batch                | 64           |
| Ratio de aprendizaje | 1e-3         |
| Optimizador          | ADAM-AMSGRAD |
| Epochs               | 100          |
| Ratio dropout        | 40%          |

Fuente: Elaboración propia. . Distribución de hiper- parámetros para imágenes 1C

- **Data set Imágenes-3C:**

*Tabla 18: Configuración para modelos en data set imágenes-3C.*

| Hiper-parámetro      | Valor        |
|----------------------|--------------|
| Batch                | 96           |
| Ratio de aprendizaje | 5e-4         |
| Optimizador          | ADAM-AMSGRAD |
| Epochs               | 100          |
| Ratio dropout        | 75%          |

Fuente: Elaboración propia. Distribución de hiper- parámetros para imágenes 3C

*Tabla 19: Aumentaciones usadas en data set imágenes-3C.*

| Aumentación          | Valor     |
|----------------------|-----------|
| CenterCrop           | 224 x 224 |
| RandomHorizontalFlip | 50%       |
| RandomVerticalFlip   | 25%       |
| RandomAffine         | 1 ~ 5     |

Fuente: Elaboración propia. Configuración de aumentos en el data set imágenes 3C